FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BİLGİSAYAR AĞLARINDA VERİ TRAFİK AKIŞININ ANALİZİ

YÜKSEK LİSANS TEZİ

ERDAL CAN YALÇIN

(150221004)

Anabilim Dalı: Bilgisayar Mühendisliği

Tez Danışmanı: Prof.Dr.Ali Yılmaz ÇAMURCU

Teslim Tarihi: 18 Haziran 2019

ONAY SAYFASI

Fatih Sultan Mehmet Vakıf Üniversitesi, Lisansüstü Eğitim Enstitüsü'nün 150221004 numaralı Bilgisayar Mühendisliği Yüksek Lisans öğrencisi Erdal Can YALÇIN,ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "BİLGİSAYAR AĞLARINDA VERİ TRAFİK AKIŞININ ANALİZİ" başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Prof. Dr. A. Yılmaz ÇAMURCU** .....
Fatih Sultan Mehmet Vakıf Üniversitesi

Jüri Üyeleri : **Prof. Dr. Burhanettin CAN** .....
Fatih Sultan Mehmet Vakıf Üniversitesi

Dr. Öğr. Üyesi Buket DOĞAN .....
Marmara Üniversitesi

Teslim Tarihi : 16 Mayıs 2019

Savunma Tarihi : 18 Haziran 2019

ÖNSÖZ

Bu tez çalışmasında desteğini esirgemeyen eşim Gamze YALÇIN'a, aileme ve çalışmanın planlanmasında, araştırılmasında, oluşumunda ilgi ve desteğini esirgemeyen, bilgi ve tecrübelerinden yararlandığım, çalışmamı bilimsel temeller ışığında şekillendiren sayın hocam Prof. Dr. Ali Yılmaz ÇAMURCU'ya sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

BİLGİSAYAR AĞLARINDA VERİ TRAFİK AKIŞININ ANALİZİ

ÖZET

Büyük kurumsal firmalar gelişimlerini teknolojiye bağlı kalmalarıyla açıklamaktadırlar. Her dönemde bir önceki dönemin teknolojisi kullanılıp bir sonraki dönemde yeni atılımlar içinde olmaları, firmaların büyümelerinin önünü açmaktadır. Şuan dünyanın en büyük firmaları/yapıları, büyümelerinin sebebini insanların kullandıkları bilgileri öğrenmeleri ile açıklamaktadırlar. Bunun en iyi örneği her bireyin bilgisayarındaki tarayıcılardaki arama motorlarıdır. Bu tarayıcılarda insanların ilgi,istek,tutumlarının ve düşüncelerinin neler olduğu tespit edilerek,sürekli bu veriler saklanmaktadır. Bu verilerle elde edilen bilginin önemi ve özellikle ileride bu bilgilerin hangi yapılara dönüşeceği önceden bilinmesi, teknolojiyi kullanan firmaların/yapıların önem verdiği alanlardandır. Burada bilginin ileride neye dönüşeceğini,insanlar arasındaki ağın ne şekilde olduğunu ve zamanla ihtiyaçların belirlenmesi için verilerin arasındaki ilişkinin tespiti daha da önem kazanacaktır. Tespit edilen verilerle insanlar teknolojiyi tahmin edip gerekli durumlarda önlemler alabilecek ve kendilerini ileride ki değişime göre güncelleme fırsatı bulacaklardır.

Yapılan tezde İstanbul Ayvansaray Üniversitesi, İnternet ve Ağ Teknolojileri 01.03.2018 ile 10.05.2018 tarihleri arasında (9 hafta) öğrencinin kullandığı laboratuvarda ders içeriğinde kullanılan verilerin akışı kontrol edilmiştir. Öğrencilerin kullandıkları bilgisayarların bir dönem boyunca haftanın aynı gününde ve aynı saatinde kullanımlarının analizi yapılmıştır. Yapılan analiz sonucunda öğrencilerin gittikleri web siteleri ayıklanıp belli sonuçlara ulaşılmıştır.

Ağ üzerinden geçen data trafiği izlenmiş ve öğrenci bazlı gidilen websitelerinin hangi oranda gittiklerini birliktelik kurallarını kullanarak tespit edilmiştir. Kullanılan birliktelik kurallarına en uygun veri olması sebebiyle Apriori algoritması ile incelenmiştir. Eclat algoritmasıyla karşılaştırılmış ve Anaconda derleyicisiyle analizi yapılmıştır. R studio ile görselleştirilmiştir. Öğrencilerin yaş, not, cinsiyet gibi değişkenlerinin de analize dahil edilerek sonuçlara etkisi gözlenmiştir. Bu veri kümesiyle gidilen websitelerin birbiriyle ilişkileri ele alınmış ve izlenen ağın analizi yapılarak, kullanılan yöntemin sonuçları belirtilmiştir.

ANALYSIS OF DATA TRAFFIC FLOW IN COMPUTER NETWORKS

ABSTRACT

Large companies explain their development with technology. The use of the technology of the previous period in each period and the new breakthroughs for the next period pave the way for companies to grow. The biggest companies / structures of the world are explaining the reason of their growth by learning the information that people use. The best example of this is the search engines in browsers on each individual's computer. In these scanners, it is determined that people's interests, wishes, attitudes and thoughts are determined and these data are kept constantly. The importance of the information obtained with these data and the fact that it is known in advance that this information will be transformed into the structures, is one of the top where the companies / structures using technology are important. Here, it will become even more important to determine what the information will turn into in the future, how the network is between people and the relationship between the data to determine the needs over time. With the data identified, people will be able to estimate technology and take measures where necessary, and they will have the opportunity to update themselves according to future changes.

In the thesis, the flow of data used in the course content was checked in the laboratory used by the student between the dates of 01.03.2018 and 10.05.2018 (9 weeks). The computers used by the students were analyzed during the same day of the week and at the same time. As a result of the analysis, the websites that students went to were detected and certain results were reached.

The data traffic over the network was monitored and the frequency at which web sites were visited was determined by using the association rules. Because it is the most suitable data for the association rules, it has been examined with Apriori algorithm, compared with Eclat algorithm and analyzed with Anaconda compiler and visualized with R studio. The effect of variables such as age, grade and gender on the results were also included in the analysis. The relationships of the websites visited with this data set were discussed and the monitored network was analyzed and the results of the method used were specified.

SİMGELER VE KISALTMALAR LİSTESİ

Simge	Açıklama
T	Tüm işlemlerin kümesi
I	Öğe kümesi
X	Öğe
$\subset$	Alt küme
$\cap$	Kesişim
$\cup$	Birleşim
$\emptyset$	Boş küme
D	İşlem kümesi
s(X)	Destek sayısı
$A \Rightarrow C$	Birliktelik Kuralı
F_k	k Uzunluğundaki Tüm Sık Öğe Kümeleri
C	Sık Kapalı Öğe Kümesi
I/O	Giriş / Çıkış
IN	İşlem tanımlayıcı listesi
L_k	Sık Geçen k Uzunluktaki Öğe kKümesi
L	Sık Öğe Küme
$K_{mak.}$	Maksimum Kapalı Sık Öğe Kümesi

Kısaltma	Açıklama
AIS	Agrawal, Imielinski & Swami
CRISP-DM	Cross Industry Standart Process for Data Mining
AI	Artificial Intelligence
MAC	Media Access Control
IP	Internet Protocol
CSV	Comma Separated Values
GEN	Cinsiyet
ORT	Mezuniyet Ortalaması
ECLAT	Equivalence Class Transformation
TID	Transaction ID (İşlem Numarası Listesi)

ŞEKİL LİSTESİ

TABLO LİSTESİ

1. GİRİŞ

Günümüzde teknolojinin ilerlemesiyle bilgisayar kullanımı her alanda değişmez bir yere sahip olmuştur. Bilgisayar kullanımın artması dünya çapında bilginin değerinin arttırmış ve bilgiye erişimin,paylaşmanın ve iletmenin kolaylaşması sağlanmıştır. Oluşan bu yapı bilginin her geçen gün daha artmasını ve bilgisayar ağının büyümesine sebep olmuştur. Dünyada bu ağın ilk akla gelen örneği internet olarak bilinmektedir. İnternetin büyümesi zamanla iletişimin kolaylaşmasına sebep olmuş ve bireyler bilgiye bu sayede daha rahat ulaşmışlardır. Bu bilgi akışı birbirini tetiklemiş ve oluşan yapı bilgisayar ağlarının analizini daha önemli hale getirmiştir. Dünyada bir çok firma, kuruluş ve devlet gelişimlerini bu ağın gösterdiği sonuçlara ve gelecekte olabilecek yapılara göre planlamaktadır.

Bu yapılar geleceğin nasıl olacağına örnek teşkil ederken bireyler iletişime geçme konusunda her geçen gün daha fazla ihtiyaç duyacaklardır. Bu bilgi akışının belli noktada getirebileceği olumsuz durumlarında oluşabileceği malümdür. Bu noktada erişilmeye çalışılan verinin kullanılan ağ üzerinde istem dışı kullanımları veya kullanılmaması gereken yerlerde, kullanılması sorununu ortaya çıkaracaktır. Verinin ağ üzerinden iletiminin nasıl olduğu ve bu iletimin analizi sonucunda verinin hangi yöne gittiğini bilmek çok önem kazanmıştır.

Bireylerin teknolojiyi kullanırken bazen istemli bazen de istemsiz olarak farklı bilgiye ulaştıkları görülmektedirler. İnsanların veriye ulaşırken ilerleyen süreçte bu verilerden ne tip sonuçlara ulaşacakları konusu, firmaların/yapıların ilgisini çekmektedir. Bu verileri elde etmenin yolu insanların davranış ve düşüncelerinin yönelimini bilmekten geçer. Bu sebeple kurumlar hatta ülkeler kendi arama motorlarının kullanılmasını teşvik etmektedir. Böylece insanların neye ulaşmak istediğini öğrenmeye çalışırlar. Bu sayede toplanan bilgilerin tek başlarına anlamlı bir ifade üretmesi beklenemez. Bir sonucun çıkarılıp geleceğe dair uygulamaya dönüşmesi için toplanan verilerin kendi aralarındaki ilişkilerinin analiz edilerek, sonuç çıkarılması gerekmektedir.

1.1.Tezin Amacı

Çalışmamızda aktif kullanılan bir laboratuvarda öğrencilerin ders esnasında kullandıkları bilgisayarların veri trafiği izlenmiştir. Öğrencilere ait MAC adresleri yardımıyla ziyaret ettikleri web siteleri esnasındaki verinin toplanması amaçlanmıştır. Toplanan veri seti ile öğrencilerin 9 haftalık ders periyodunda hangi sitelere gittikleri ve öğrencilerin websitesi alışkanlıklarının tespit edilmesi hedeflenmiştir. Öğrencilerin veri trafik akışının izlendiği dersten aldıkları not ağırlığına ve cinsiyetlerine göre tercih ettikleri websiteler tespit edilip bu websitelerin arasında birliktelik ilişkisinin olup olmadığının analizi yapılmıştır. Farklı çalışmalarda bireylerin kullandığı websitelerin yoğunluğu ve tercih edilmeleri ile ilgili çalışmalar bulunmaktadır. Bu çalışmada farklı olarak gidilen web sitelerin birbirleriyle ilişkilerinin analizi yapılmıştır. Böylece potansiyel tehdit oluşturabilecek veya olumlu yönlendirmeler sağlayabilecek websitelerinde tespit edilmesi amaçlanmaktadır. Apriori algoritmasıyla yapılan ilişki analizinin Eclat algoritmasıyla sağlaması yapılmıştır. Yapılan sağlama ile birliktelik ilişkisinin doğruluğunun güçlenmesi hedeflenmiştir. Aynı zamanda analiz sonuçlarından çıkarımlarda bulunup öneriler sunulmuştur.

1.2. Tezin Yapısı

Yapılan çalışmada beş ana bölüm yer almaktadır.Bu ana bölümlere bağlı alt bölümlerle konular bütünleştirilmiştir. Giriş bölümünde tezin amaç ve yapısı değerlendirilmiştir. İkinci bölümde literatür taraması yapılarak birliktelik kuralları tanımı, madenciliği,matematiksel modeli ve temel kavramlardan söz edilmiştir. Ayrıca bu bölümde genel olarak Apriori ve Eclat algoritmasının temel mantığı anlatılmıştır. Üçüncü bölümde kullanılan sistemin tasarımı ve veri madenciliği yaşam döngüsünden yola çıkarak tezin yapısı anlatılmıştır. Dördüncü bölümde ağdan aldığımız log kayıtlarından verinin anlamlı bir yapıya dönüştürülmesi için geçen sürecin uygulama adımları görsellerle desteklenerek açıklanmıştır. Bu bölümde anaconda derleyicisi üzerinde verinin analizi yapılmıştır. Yapılan analizlerle çıkan sonuçlar görselleştirilmiştir. Apriori ve Eclat algoritmaları, R Studio derleyici üzerindeki kütüphaneler yardımıyla çalıştırılmış ve analiz sonuçları bu bölümde gösterilmiştir. Beşinci bölüm sonuç bölümüdür. Bu bölümde yapılan analizler değerlendirilmiş ve elde edilen tespitler yazılmıştır. Ayrıca analiz sonuçlarına göre önerilerde bulunulmuştur.

2. LİTERATÜR TARAMASI

Birliktelik kuralları, birbiriyle ilişkili olan özelliklerin ortaya çıkarılması ve aralarındaki ilişkinin büyüklüğünün tespit edilmesini amacıyla kullanılan kurallar bütünüdür.

2.1. Birliktelik Kuralları (Association Rules) Tanımı

Birliktelik kurallarının arkasındaki kavramlar daha erken izlenebilse de, 1990'lı yıllarda birliktelik kural madenciliği tanımlandı, bilgisayar bilimcileri Rakesh Agrawal, Tomasz Imieliński ve Arun Swami'nin satış noktasını kullanan ürünler arasındaki ilişkileri bulmak için algoritma tabanlı bir yöntem geliştirdiler (POS sistemleri). Algoritmaları süpermarketlere uygulayan bilim adamları, satın alınan ürünlerde, birleşme kuralları olarak adlandırılan farklı öğeler arasındaki bağlantıları keşfettiler ve sonuçta bu bilgileri, farklı ürünlerin birlikte satın alınma ihtimalini tahmin etmek için kullandılar.

İlişkilendirme kuralları, çeşitli veri tabanlarındaki büyük veri kümelerindeki veri öğeleri arasındaki ilişkilerin olasılığını göstermeye yardımcı olan if-then ifadeleridir. Birliktelik kuralı madenciliğinde çok sayıda uygulama vardır ve işlem verilerinde veya tıbbi veri kümelerinde satış ilişkilerini keşfetmeye yardımcı olmak için yaygın olarak kullanılır. Topluluk kuralı madenciliği, temel düzeyde, bir veritabanındaki modeller veya eş-oluşumlar için verilerin analizinde makine öğrenme modellerinin kullanılmasını içerir. Bir ilişkilendirme kuralı iki bölümden oluşur: bir öncül (eğer) ve bir sonuç (sonra). Bir öncül veri içinde bulunan bir maddedir. Bunun bir sonucu, öncül ile birlikte bulunan bir maddedir. [1]

Birliktelik kuralları, sık rastlanan kalıplar için veri arayarak ve en önemli ilişkileri tanımlamak için destek ve güven ölçütlerini kullanarak oluşturulur. Destek, öğelerin verilerde ne sıklıkta göründüğünün bir göstergesidir. Güven, if-then ifadelerinin doğru bulunma sayısını gösterir. Güven ile beklenen güven arasında karşılaştırma yapmak için, asansör adı verilen üçüncü bir ölçüm kullanılabilir.

Birliktelik kuralları, iki veya daha fazla maddeden oluşan öğe kümelerinden hesaplanır. Kurallar, tüm olası öğe kümelerini analiz etmek için kurulursa, kuralların çok az anlam taşıdığı birçok kural olabilir. Bununla beraber, ilişkilendirme kuralları tipik olarak verilerde iyi temsil edilen kurallardan oluşturulur.

Agrawal, Imielinski & Swami (AIS) algoritması ile öğe kümeleri üretilir ve veriler tararken sayılır. İşlem verilerinde, AIS algoritması, hangi büyük öğelerin bir işlem içerdiğini belirler ve büyük öğe kümeleri, işlem verilerindeki diğer öğelerle genişletilerek yeni aday öğeler oluşturulur.

SETM algoritması bir veritabanını tararken aday öğeler oluşturur, ancak bu algoritma taramanın sonunda bulunan öğeler için de geçerlidir. Yeni aday öğe setleri, AIS algoritması ile aynı şekilde üretilir, ancak üretici işlemin işlem kimliği, aday öğe kümesi ile sıralı bir yapıda kaydedilir. Geçişin sonunda, aday öğe kümelerinin destek sayısı, sıralı yapının toplanmasıyla yaratılır. Hem AIS hem de SETM algoritmalarının dezavantajı, Real Time Data Mining'in yazarı Dr. Saed Sayad'ın yayınladığı materyallere göre, her birinin birçok küçük aday ürün kümesi üretip yapabilmesidir. [1]

Apriori algoritması ile aday öğe kümeleri, önceki geçişte yalnızca büyük öğe kümeleri kullanılarak üretilir. Bir önceki geçişteki büyük öğe kümesi, bir boyut daha büyük olan tüm öğe kümelerini oluşturmak için kendisiyle birleştirilir. Oluşturulan her öğe, büyük olmayan bir alt kümeye sahip olarak silinir. Apriori algoritması, sık bir öğe kümesinin herhangi bir alt kümesini de sık bir öğe kümesi olarak kabul eder. Bu yaklaşımla, algoritma, Sayad'a göre, yalnızca destek sayısı minimum destek sayısından büyük olan ögeleri belirleyerek değerlendirilen aday sayısını azaltır.

Veri madenciliğinde, birleşme kuralları müşteri davranışını analiz etmek ve tahmin etmek için kullanışlıdır. Müşteri analitiği, pazar sepeti analizi, ürün kümelemesi, katalog tasarımı ve mağaza düzeninde önemli bir rol oynarlar.

Programcılar, makine öğrenmesi için yetenekli programlar oluşturmak için ilişkilendirme kurallarını kullanır. Makine öğrenmesi, açıkça programlamadan daha verimli hale gelebilecek becerilere sahip programlar oluşturmayı amaçlayan bir yapay zeka türüdür.

Klasik bir birliktelik kuralı madenciliği örneği çocuk bezi ve kolalar arasındaki ilişkiyi ifade eder. Kurgusal gibi görünen örnek, çocuk bezi almak için bir mağazaya giden erkeklerin de kola alabileceklerini iddia ediyor. Buna işaret edecek veriler şöyle görünebilir:

Bir süpermarkette 200.000 müşteri işlemi vardır. Yaklaşık 4.000 işlem veya toplam işlemlerin yaklaşık% 2'si, çocuk bezinin satın alınmasını içerir. Yaklaşık 5.500 işlem (% 2.75) kola alımını içermektedir. Bunların yaklaşık 3.500 işlemi, % 1.75'i, çocuk bezi ve kola alımını içermektedir. Yüzdelere göre, bu sayı daha düşük olmalıdır. Bununla birlikte, bebek bezi alımlarının yaklaşık% 87,5'inin kola alımı içermesi gerçeği, çocuk bezi ve kola arasında bir bağlantı olduğunu gösterir.

Birçok işletme, günlük işlemlerinden günlük olarak büyük miktarlarda veri toplar. Örneğin, marketlerin kasalarında günlük olarak büyük miktarlarda müşteri alım verileri toplanmaktadır. Tablo 2.1'de, genel olarak pazar sepeti işlemleri olarak bilinen bunun bir örneğini göstermektedir. Bu tablodaki her satır, İşlem Numarası Listesi (TID) etiketli benzersiz bir tanımlayıcı ve belirli bir müşteri tarafından satın alınan bir ürün kümesi içeren bir işleme karşılık gelir. Perakendeciler, müşterilerinin satın alma davranışları hakkında bilgi edinmek için verileri analiz etmek ile ilgilenmektedir. Bu değerli bilgiler, pazarlama promosyonları, envanter yönetimi ve müşteri ilişkileri yönetimi gibi işle ilgili çeşitli uygulamaları desteklemek için kullanılabilir.

Tablo 2.1: Örnek bir piyasa taslağı düzenlemesi [1]

İşlem Numarası Listesi (TID)	*Öğeler*			
1	*Ekmek*	*Süt*		
2	*Ekmek*	*Bebek Bezi*	*Kola*	*Yumurta*
3	*Süt*	*Bebek Bezi*	*Kola*	*Kola*
4	*Ekmek*	*Süt*	*Bebek Bezi*	*Kola*
5	*Ekmek*	*Süt*	*Bebek Bezi*	*Kola*

Tablo 2.1'de gösterilen verilerden yandaki kural çıkarılabilir: Bebek Bezi ⇒ Kola
Kural, çocuk bezi ve kola satışı arasında güçlü bir ilişki olduğunu gösteriyor çünkü çocuk bezi satın alan birçok müşteri de kola alıyor. Perakendeciler, ürünlerini müşterilere çaprazlamak için yeni fırsatları belirlemelerine yardımcı olmak için bu tür araçları kullanabilir.
Pazar sepeti verilerinin yanı sıra, birlik analizi de biyoinformatik, tıbbi teşhis, Web madenciliği ve bilimsel veri analizi gibi diğer uygulama alanlarına uygulanabilir. Örneğin, Dünya bilimi verilerinin analizinde, ilişkilendirme modelleri okyanus, kara ve atmosferik süreçler arasındaki ilginç bağlantıları ortaya çıkarabilir. Bu tür bilgiler, dünya bilim adamlarının, dünya sisteminin farklı unsurlarının birbirleriyle nasıl etkileşime girdiğini daha iyi anlamalarına yardımcı olabilir. Burada sunulan teknikler genellikle daha geniş bir yelpazedeki veri kümelerine uygulanabilir olsa da, açıklama amaçlı olarak tartışmamız temel olarak pazar sepeti verilerine odaklanacaktır.
Piyasa sepeti verilerine ilişki analizi uygulanırken ele alınması gereken iki önemli konu vardır. İlk olarak, büyük bir işlem veri kümesinden kalıpları keşfetmek hesaplama açısından pahalı olabilir. İkincisi, keşfedilen modellerden bazıları potansiyel olarak sahtedir, çünkü bunlar sadece tesadüfen olabilir.

İkili Gösterim Piyasa sepeti verilerinde, her satır bir işleme ve her sütun bir öğeye karşılık gelir. Tablo 2.2'deki gibi ikili biçimde gösterilir. Bir öğe, bir işlemde mevcut değeri sıfır, aksi halde sıfır olan bir ikili değişken olarak değerlendirilebilir. Bir öğenin varlığı, çoğunlukla bulunmamasından daha önemli olarak kabul edildiğinden, bir öğe asimetrik bir ikili değişkendir.

Tablo 2.2: İkili temsil sepeti verisi [1]

İşlem Numarası Listesi(TID)	*Ekmek*	*Süt*	*Bebek Bezi*	*Kola*	*Yumurta*	*Cola*
1	*1*	*1*	*0*	*0*	*0*	*0*
2	*1*	*0*	*1*	*1*	*1*	*0*
3	*0*	*1*	*1*	*1*	*0*	*1*
4	*1*	*1*	*1*	*1*	*0*	*0*
5	*1*	*1*	*1*	*0*	*0*	*1*

Bu temsil gerçek piyasa sepeti verilerinin çok basit bir görünümüdür, çünkü verilerin satılan ürün miktarı veya onları satın almak için ödenen fiyat gibi belirli önemli yönlerini göz ardı eder.

Öğe Kümesi ve Destek Sayısı: I = {i1, i2,...,id} bir pazar sepetindeki tüm öğelerin kümesi ve T = {t1, t2,...,tn } tüm işlemlerin kümesi olsun.
Her işlem ti, I'den seçilen öğelerin bir alt kümesini içerir. İlişkilendirme analizinde, sıfır veya daha fazla maddeden oluşan bir koleksiyon öğekümesi olarak adlandırılır. Bir öğe kümesi k öğeleri içeriyorsa, buna k-öğe kümesi (k-itemset) adı verilir. {Kola, Çocuk Bezi, Süt} bir k-öğe kümesi örneğidir. Boş küme, herhangi bir madde içermeyen bir demettir. [1]
İşlem genişliği, bir işlemde bulunan kalemlerin sayısı olarak tanımlanır. X in Tj nin bir alt kümesi olması durumunda, bir işlem tj nin X öğe setini içerdiği söylenir.

Örneğin, Tablo 2.2`de gösterilen ikinci işlem {Ekmek, Bebek Bezi} öğe kümesini içerir, ancak {Ekmek, Süt} öğesini içermez. Bir öğe kümesi'nin önemli bir özelliği, belirli bir öğe kümesi içeren işlemlerin sayısını ifade eden destek sayımıdır. Matematiksel olarak, bir öğe kümesi için *σ(X)* destek sayısı X olarak belirtilebilir:

$$\sigma(X) = |\{t_i | X \subseteq t_i , t_i \in T\}|, \qquad (2.1)$$

Tablo 2.2 de gösterildiği gibi {Kola, Bebek Bezi, Süt} için destek sayısı ikiye eşittir, çünkü üç öğenin tümünü içeren yalnızca iki işlem vardır.

Genellikle ilgilenilen özellik, bir öğe kümesinin gerçekleştiği işlemlerin kesri olan destekdir:

$$s(X) = \sigma(X) / N. \qquad (2.2)$$

s(X) kullanıcı tarafından tanımlanan bazı eşik(minsup) değerlerden daha büyükse, X öğe kümesine sık denir.

2.2. Birliktelik Kuralı Madenciliği

Birliktelik kuralı madenciliği, ilişkisel veri tabanları, işlem veri tabanları ve diğer veri depolama araçları gibi çeşitli veri tabanlarında bulunan veri kümelerinden sık desenler, korelasyonlar, birliktelikler veya nedensel yapılar bulmak için kullanılan bir prosedürdür. [2]

Bir dizi işlem göz önüne alındığında, birliktelik kuralı madenciliği, işlemdeki diğer öğelerin oluşumlarına dayanarak belirli bir öğenin oluşumunu tahmin etmemizi sağlayan kuralları bulmayı amaçlamaktadır.

Birliktelik kuralı madenciliği, öğe kümeleri arasındaki birliktelikleri ve nedensel nesneleri yönetebilecek kuralları bulan bir veri madenciliği sürecidir. Bu nedenle, birden fazla öğe ile verilen belirli bir işlemde, bu öğelerin nasıl ve neden sıklıkla birlikte yer aldığını düzenleyen kuralları bulmaya çalışır. Örneğin, şaşırtıcı bir şekilde, çocuk bezleri ve kola birlikte satın alınmaktadır, çünkü anneler bebekle birlikte kalırken babalar genellikle alışveriş yapmakla görevlidir.

2.3. Birliktelik Kuralları (Association Rules) Matematiksel Modeli ve Temel Kavramları

Birliktelik kuralı öğrenme, büyük veritabanlarındaki değişkenler arasındaki ilginç ilişkileri keşfetmek için kurala dayalı bir makine öğrenme yöntemidir. Bazı ilginç ölçütleri kullanarak veri tabanlarında keşfedilen güçlü kuralları belirlemek amaçlanmıştır. Nihai hedef, yeterince büyük bir veri kümesini varsayarak, bir makinenin insan beyninin, kategorize edilmemiş yeni verilerden çıkarılması ve soyut birleştirme yeteneklerini taklit etmesine yardımcı olmaktır.
Bir olası yapı, tüm olası kalem setlerinin listesini numaralandırmak için kullanılabilir.

2.3.1. Birliktelik kuralları (association rules) matematiksel modeli

Birliktelik kuralının matematiksel modeli 1993 yılında Agrawal, Imielinski ve Swami tarafından keşfedilmiştir. Modelde, nesneler kümesi $I = \{i_1, i_2, i_3, \ldots\ldots, i_m\}$ ve işlem kümesi D olarak ifade edilir. Her "i" farklı bir nesneye (ürün) karşılık gelir. D veri tabanındaki her işlem, $T \subset I$ olarak tanımlanan bir öğe kümesidir. IN, her işlem için benzersiz bir sayıdır ve m öğe sayısıdır. A ve B, nesne kümelerini temsil eder. Eğer ve sadece $A \subset T$ ise T işlemleri dizisinin A içerdiği söylenir. Yani eğer A, T nin bir alt kümesi ise. [2]

A ve B'nin koşulları yerine getirmek için öğeler olduğunu varsayalım. Bu durumda,

$A \subset I$, $B \subset I$ ve $A \cap B = \emptyset$ olur. [3]

2.3.2. Destek(support) ve güven(Confidence) değeri

Verilerdeki desenleri analiz ederken, gerçekten aranan şey ilginç olan desenlerdir. Verilerin ilginç olup olmadığını belirlemenin öznel yolları vardır, ancak "ilginçlik" için objektif önlemler oluşturarak veri analizi hızlandırabilir. Verilerdeki kalıpları, ilişkileri ve korelasyonları ararken, birçok algoritma objektif destek (2.4) ve güven (2.5) ölçütlerini kullanır. [4] [5]

$$destek(A \rightarrow C) = destek(A \cup C) \qquad (2.4)$$

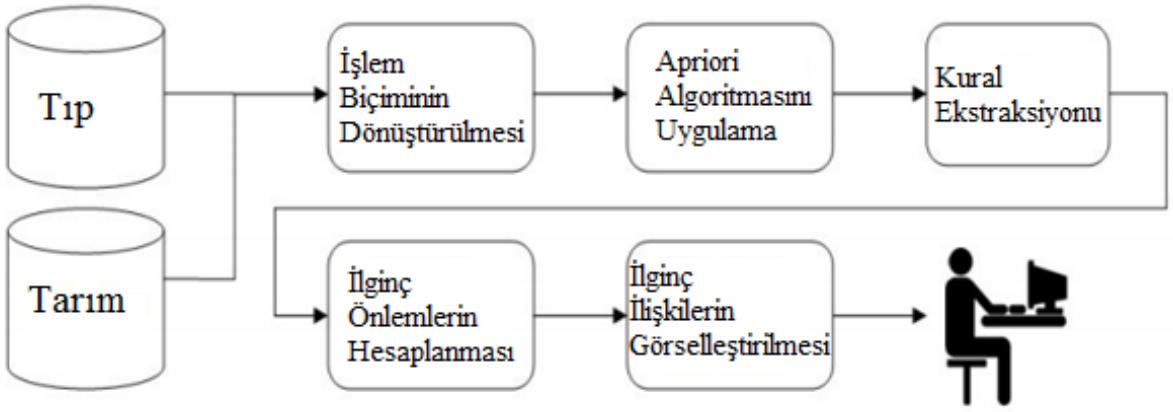
Destek ölçütü, ilişkilendirme kuralları için değil, öğe kümeleri için tanımlanmıştır. Birliktelik kuralı madenciliği algoritması tarafından üretilen tablo üç farklı destek ölçütü içerir: 'öncül destek', 'sonuç destek' ve 'destek'. Öncül destek', öncül A'yı içeren işlemlerin oranını hesaplar ve 'sonuçtaki destek', sonuçtaki C öğesinin desteğini hesaplar. 'Destek' ölçümü daha sonra birleşik öğe kümesi A ∪ C'nin desteğini hesaplar.

Tipik olarak, destek bir veri tabanındaki bir öğe kümesinin bolluğunu veya sıklığını ölçmek için kullanılır. Bu, bir öğe kümesinin sıklığının öğe kümesi sayısına ve veri kümesinin boyutuna bağlı olduğu anlamına gelir. Az sayıda öğe setinin destek değeri, aynı öğeler için daha büyük bir öğe grubuna sahip bir veri kümesine kıyasla daha yüksektir. [3]

$$güven(A \rightarrow C) = \frac{destek(A \rightarrow C)}{destek(A)} \qquad (2.5)$$

Güven, kuralın ne kadar sıklıkla doğru olduğunun bir göstergesidir. A ⇒ C kuralının güvenirliği, öncülü de içerdiği için yapılan bir işlemde sonucu görme ihtimalini ifade eder.
A ⇒ C'ye olan güven C ⇒ A'ya olan güvenden farklıdır. Sonuç ve öncül her zaman birlikte ortaya çıkarsa, A ⇒ C kuralı için güven 1'dir (maksimal). [3]

Ortak eşik problemi, kullanıcı tarafından belirlenen minimum desteği ve güveni sağlayan tüm birliktelik kurallarını bulmaktır. Yüksek minimum destek seviyesinde, sadece birkaç kural oluşturmak ve veritabanını taramak için daha fazla zaman harcamak gerekebilir. Düşük asgari destek değeri seçilirse çok sayıda gereksiz kural üretebilir. Bu nedenle, bazı istatistiksel ilginçlik ölçütleri kullanılarak sorun giderilir. İstatistiksel bir ölçü, veri madenciliği uygulamalarına anlamlı matematiksel işlev türetmekten başka bir şey değildir. Kuralların ilginç mi yoksa ilgi çekici mi olduğu doğru bir şekilde belirlenmelidir. İlginçlik ölçütleri kullanarak ilginç kurallar bulma mimarisi de Şekil 2.1'de gösterilmektedir. [3]



Şekil 2.1: İlginçlik ölçütleri kullanarak ilginç kurallar bulma mimarisi [3]

Sıralı veri tabanı seçimi, işlem formatı dönüşümü, apriori algoritması uygulaması, kural çıkarma, ilginç kural hesaplama ve son olarak bu ilginç kuralları görselleştirmede altı adım işlemden geçmelidir.

- Asansör(Lift)(2.6) : Bu ölçüt, genel olarak A ⇒ C kuralının öncülünün ve sonucunun, istatistiksel olarak bağımsız olsaydı beklediğimizden daha ne kadar sıklıkla meydana geldiğini ölçmek için kullanılır. A ve C bağımsızsa, bu değer tam olarak 1 olacaktır. [4] Öncül ve sonucun bağımsız olup olmadığını gösteren oransal destek değeridir. Asansör değerleri pozitif veya negatif korelasyonlu olabilmektedir. Hesaplanan değerin 1'den ne kadar büyükse o derecede pozitif korelasyon olduğu ifade edilebilir.

$$asansör(A \rightarrow C) = \frac{güven(A \rightarrow C)}{destek(C)} \qquad (2.6)$$

2.3.3. K-nesne küme (k-itemset)

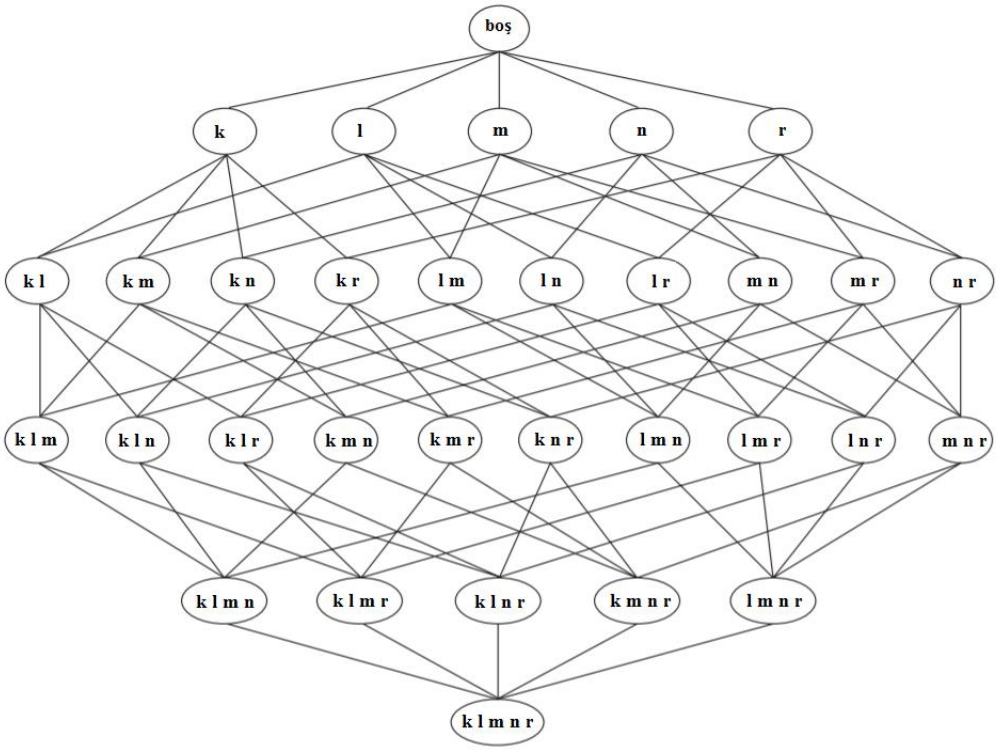
Birliktelik Madenciliği, veri setindeki sık maddeleri araştırır. Sık madencilikte genellikle işlemsel ve ilişkisel veritabanlarındaki madde kümeleri arasındaki ilginç ilişki ve korelasyonlar bulunur.

Bir küme k tane öğe içeriyorsa bu küme k-nesneküme olarak ifade edilir. İlgili destek sayısının minimum destek sayısından büyük olması durumunda bir öğe kümesinin sık olduğu söylenebilir. Sık öğe kümesi L_k şeklinde gösterilir. Örneğin; {Ekmek, Yumurta, Süt, Bal} 4 elemanlı bir nesne kümedir ve 4-nesneküme olarak ifade edilir.

2.3.4. Sık nesne kümesi (frequent itemset)

Bir kafes yapısı, tüm olası öğe setlerinin listesini numaralandırmak için kullanılabilir.

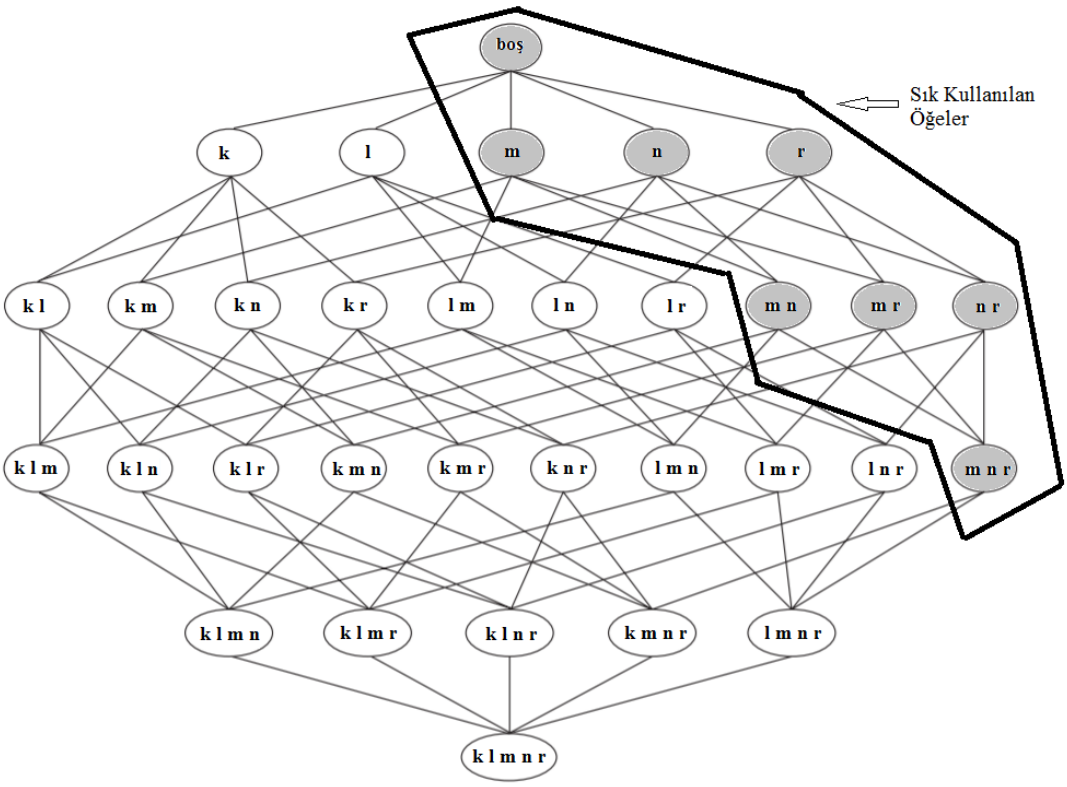
Şekil 2.2: {k,l,m,n,r } için bir öğe kümesi kafesini gösterir. Genel olarak, k öğesi içeren bir veri kümesi potansiyel olarak boş küme hariç olmak üzere $2^k - 1$ sık öğe kümesi oluşturabilir.



Şekil 2.2: Bir öğe kümesi kafes [1]

Teorem (Apriori Prensibi) : Bir öğe kümesi sıksa, o zaman tüm alt kümelerinin de sık olması gerekir.

Apriori prensibinin arkasındaki fikir Şekil 2.3'de gösterildiği gibidir. Diyelim ki {m,n,r} sık kullanılan bir öğe kümesidir. Buna göre {m,n,r} içeren herhangi bir işlemin, {m,n}, {m,r}, {n,r}, {m}, {n} ve {r} alt kümelerini de içermesi gerekir.

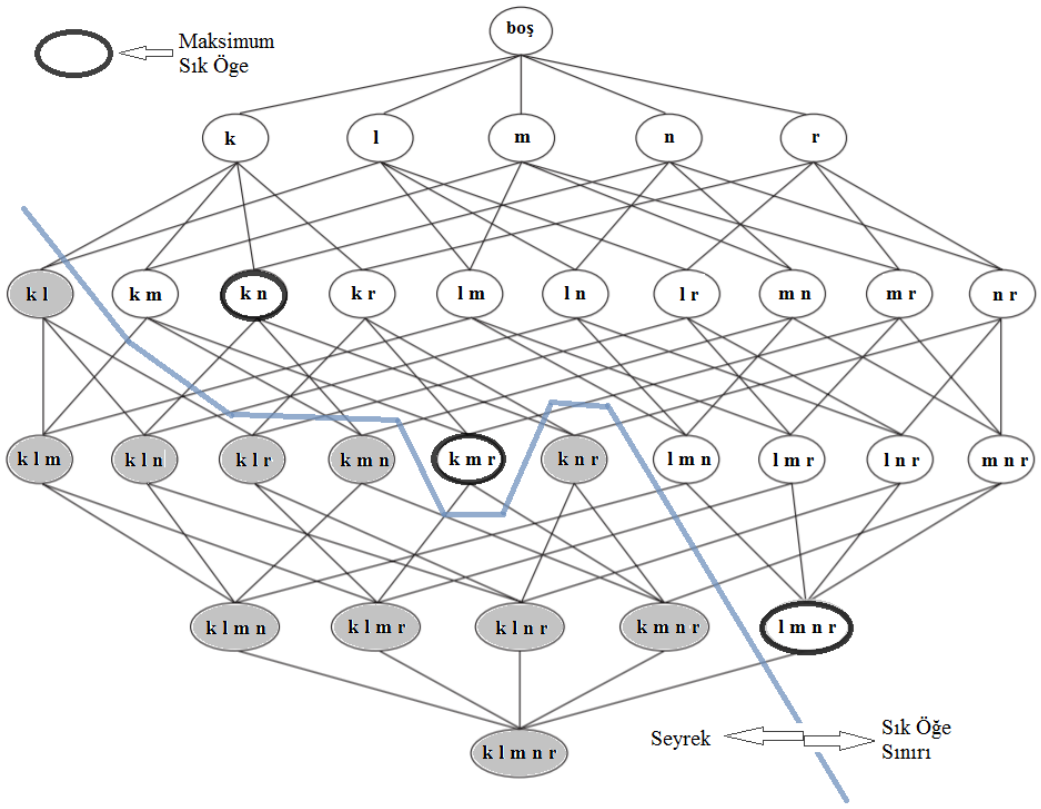


Şekil 2.3: Apriori ilkesinin {m,n,r} sık öğe örneği [1]

Buna karşılık, {k,l} gibi bir öğe kümesi nadirse, o zaman tüm üst kümelerinin de nadir olması gerekir. Şekil 2.4'de gösterildiği gibi, {k,l} 'nin üst kümelerini içeren tüm alt düğümler, {k,l}' nin nadir olduğu tespit edildikten hemen sonra budanabilir. Destek ölçüsünü temel alarak üstel arama alanını kısaltma stratejisi, destek tabanlı budama olarak bilinir. Böyle bir budama stratejisi, destek ölçüsünün kilit bir özelliği, yani bir öğe setinin desteğinin alt kümelerinin desteğini asla geçmemesi ile mümkün olmaktadır. Bu özellik, destek önleminin monoton karşıtı özelliği olarak da bilinir.

Uygulamada, bir işlem veri setinden üretilen sık öğe setlerinin sayısı çok büyük olabilir. Diğer tüm sık öğe setlerinin türetilebileceği küçük bir temsilci öğe kümesi tanımlamakta fayda vardır. Bir maksimum sıklıkta öge kümesi, yakın üst kümelerinin hiçbirinin sık olmadığı sık öğe kümesi olarak tanımlanır. Şekil 2.4'de gösterilen kafes içindeki öğe kümeleri iki gruba ayrılır: sık ve seyrek(nadir) olanlar. Ayrıca, açık mavi renkteki çizgi ile temsil edilen sık öğe kümesi sınırı da diyagramda gösterilmektedir. Sınırın üstünde bulunan her öğe kümesi sık iken sınırın altında bulunanlar (gölgeli çemberler) nadirdir. Sınırın yakınında bulunan öğeler arasında, {k, n}, {k, m, r) ve {l,m,n,r} yakın üst öğe kümeleri nadir olduğundan, en sık rastlanan öğeler olarak kabul edilir. Örneğin, {k,n} gibi bir öğe kümesi maksimum sıklıktadır çünkü {k, l, n}, {k, m, n} ve {k, n, r} nadirdir. Buna karşılık, {k, m} maksimal değildir çünkü en yakın üst kümelerinden biri olan {k, m, r} sık görülür. Maksimum sık öğe kümeleri, sık öğe kümelerinin kompakt bir gösterimini sağlar. Başka bir deyişle, tüm sık öğe kümelerinin türetilebileceği en küçük öğe kümesini oluştururlar.
Örneğin, Şekil 2.4'daki her sık öğe kümesi, üç azami sık öğe kümesinden birinin alt kümesidir, {k, n}, {k, m, r} ve {l, m, n, r}.

Bir öğe kümesi maksimum sıklıktaki öğe kümelerinin herhangi birinin uygun bir alt kümesi değilse, o zaman ya nadirdir (örneğin, {k, n, r}) ya da azami sıklıktır (örneğin, {l, m, n, r}). Dolayısıyla, {k, m, r}, {k, n} ve {l, m, n, r} maksimal sık ürün setleri, Şekil 2.4'da gösterilen sık ürün setlerinin kompakt bir gösterimini sağlar. Maksimal sık öğe setlerinin tüm alt kümelerini numaralandırmak, tüm sık öğe setlerinin tam listesini oluşturur.



Şekil 2.4: Maksimum sıklık öğe kümesi [1]

Maksimum sıklıkta ürün setleri, veride üstel olarak birçok sık öğe seti bulunduğundan, çok uzun, sık öge kümeleri üretebilen veri kümeleri için değerli bir temsil sağlar. Bununla birlikte, bu yaklaşım yalnızca, en sık kullanılan öge setlerini açıkça bulmak için etkili bir algoritma mevcutsa pratiktir.

Kompakt bir sunum sağlamasına rağmen, maksimum sıklıkta bulunan öğe setleri alt gruplarının destek bilgilerini içermez. Örneğin, {k, m, r }, {k, n} ve {l, m, n, r} azami sıklıktaki öğe kümelerinin desteği, alt kümelerinin desteği hakkında, destek eşiğini karşılaması dışında herhangi bir bilgi sağlamaz. Bu nedenle, maksimum olmayan sıklıktaki öğelerin destek sayımlarını belirlemek için veri kümesi üzerinden ek bir geçiş gereklidir. Bazı durumlarda, destek bilgisini koruyan öğe setlerinin asgari düzeyde gösterilmesi istenebilir. Kapalı öğe kümeleri bu durumu karşılamaktadır.

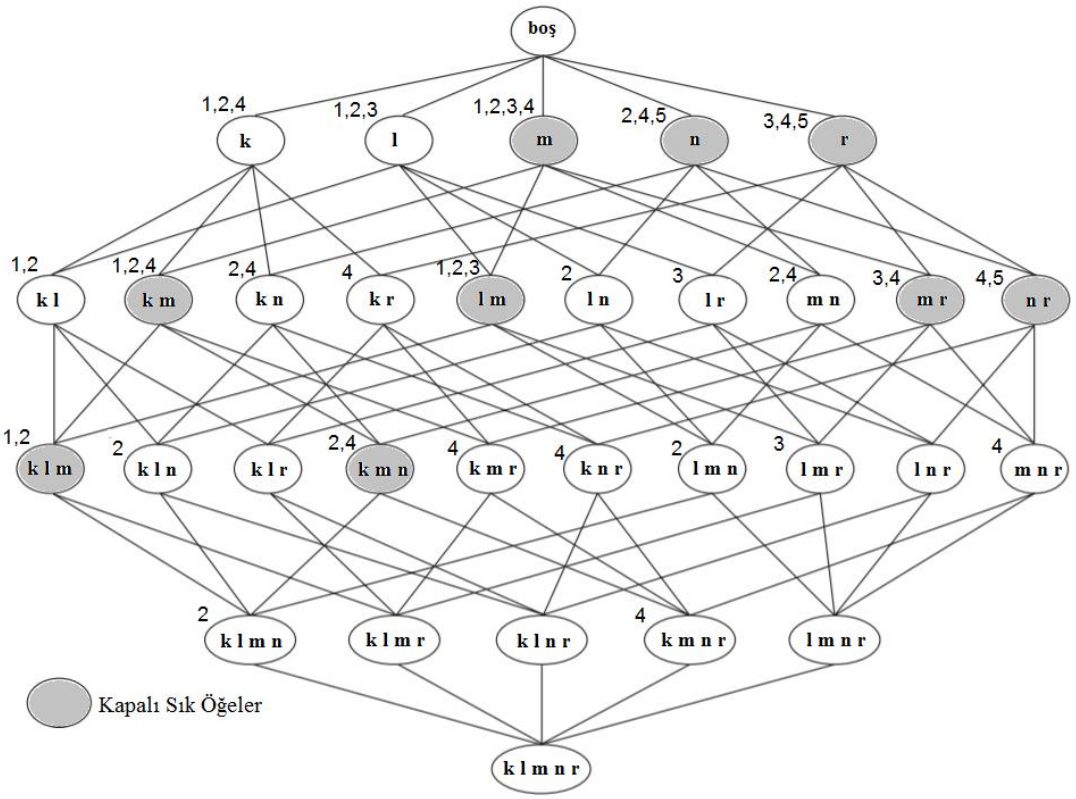
Kapalı öğe setleri, destek bilgilerini kaybetmeden tüm ürün setlerinin asgari düzeyde gösterilmesini sağlar. Örneğin, en yakın bir üst kümesinin hiçbiri A ile tam olarak aynı destek sayısına sahip değilse, A öğe kümesi kapalıdır. Başka bir deyişle, en yakın üst ögelerinden en

az biri A ile aynı destek sayısına sahipse, A kapalı değildir. Kapalı öğe kümelerinin örnekleri, Şekil 2.5'de gösterilmektedir. Her öğe grubunun destek sayısını daha iyi göstermek için, kafesdeki her düğümü (öğe kümesi) karşılık gelen işlem kimlikleri listesiyle ilişkilendirilir.

Tablo 2.3: Şekil 2.5'in verilerini içeren tablo bilgisi[1]

İşlem Numaraları Listesi (TID)	Öğeler
1	klm
2	klmn
3	lmr
4	kmnr
5	nr

**Minsup=%40*



Şekil 2.5: Sık kapalı öğe kümesi [1]

Örneğin, {l, m} düğümü 1, 2 ve 3 numaralı işlem kimlikleriyle ilişkili olduğundan, destek sayısı üçe eşittir. Bu şemada verilen işlemlerden, {l} desteği {l, m} ile aynıdır. Bunun nedeni l içeren her işlemin m içermesidir. Dolayısıyla, {l} kapalı bir öğe kümesi değildir. Benzer şekilde, hem k hem de n'yi içeren her işlemde m gerçekleştiğinden, {k, n} öğe kümesi üst kümesi {k, m, n} ile aynı desteğe sahip olduğundan kapalı değildir. Öte yandan, {l, m} kapalı bir öğedir, çünkü

üst kümelerinin hiçbiriyle aynı destek sayısına sahip değildir. Bir öğe, kapalıysa ve desteği, minsup tan büyük veya ona eşitse, kapalı bir sık öğe kümesidir.
Şekil 2.5'deki örnekte, destek eşiğinin% 40 olduğu varsayımıyla, {l, m}, kapalı bir sık öğe kümesidir, çünkü desteği% 60'tır. Ve kapalı sık ürün kümeleri gölgeli düğümlerle belirtilmiştir.

Belirli bir veri setinden kapalı sık öğe setlerini açıkça çıkarmak için algoritmalar mevcuttur. Kapalı olmayan sık öğe setlerinin destek sayısını belirlemek için kapalı sık kullanılan öğe kümelerini kullanabiliriz. Örneğin, Şekil 2.5'de gösterilen sık öğe setini {k, n} ele alınabilir. Bu öğe kümesi kapalı olmadığı için, destek sayısı, en yakın üst kümelerinin maksimum destek sayısına eşit olmalıdır. Ayrıca, {k, n} sık olduğundan, sadece sık üst öğe kümelerinin desteği göz önüne alınır. Genel olarak, kapalı olmayan her sık k-öğe kümesinin destek sayısı, k + 1 büyüklüğündeki tüm sık üst öge kümelerinin desteği dikkate alınarak elde edilir. Örneğin, {k, n} 'nin tek sık kullanılan üst kümesi {k, m, n} olduğundan, {k, m, n} desteğine eşittir (Destek = 2). Bu metodolojiyi kullanarak, her sık öğe setinin desteğini hesaplamak için bir algoritma geliştirilebilir. Bu algoritma için algoritmanın sözde kodu Şekil 2.6'te gösterilmiştir.

1: *C sık kapalı öğe setini ve F tüm sık öğe setini göstersin.*

2: $K_{mak.}$ *maksimum kapalı sık öğe setini göstersin.*

3: $F_{k_{mak.}} = \{f \mid f \in C, |f| = k_{mak.}\}$ *{$k_{mak.}$ uzunluğundaki tüm sık öğe setlerini bulunur}*

4: for $k = k_{mak} - 1$ *1 e kadar*

5: $F_k = \{f | f \in F, |f| = k\}$ *{k. uzunluğundaki tüm sık öğe setlerini bulunur}*

6: ***for each*** $f \in F_k$ ***do***

7: ***if*** $f \notin C$ ***then***

8: $f.destek = mak\{f'.destek \mid f' \in F_{k+1},\ f \subset f'\}$

9: ***end if***

10: ***end for***

11:end for

Şekil 2.6: Kapalı sık öğe setlerinin destek değerlerinin hesaplanması [1]

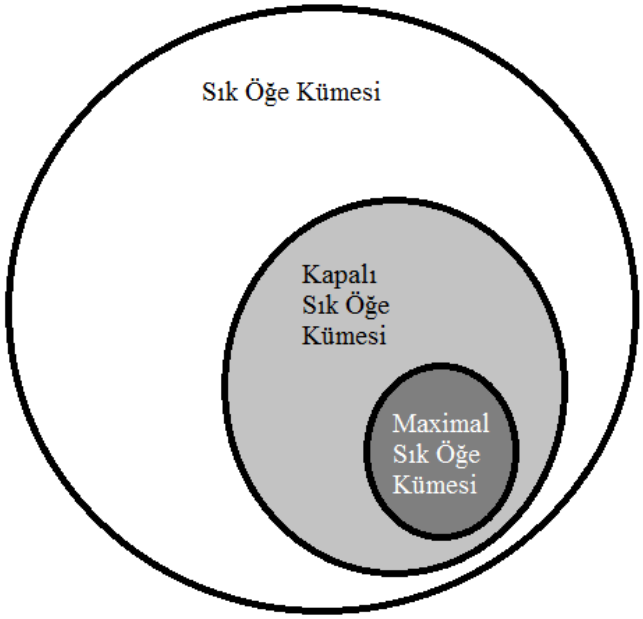
Algoritma, en büyüğünden en küçük sıklıktaki öge kümelerine ilerler. Bunun nedeni, kapatılmayan bir sık öğe setinin desteğini bulmak için, tüm üst kümelerin desteğinin bilinmesi gerekir. [6]

Sık kapalı öğe setleri kullanmanın avantajı, on adet işlem ve onbeş ürün içeren Tablo 2.4'te gösterilmiştir.

Tablo 2.4: Kapalı öğe kümelerini incelemesi için ayarlanmış bir işlem verisi. [1]

TID	k_1	k_2	k_3	k_4	k_5	l_1	l_2	l_3	l_4	l_5	m_1	m_2	m_3	m_4	m_5
1	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0
2	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0
3	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0
4	0	0	**1**	**1**	0	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0
5	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0
6	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**
8	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**
9	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**
10	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**

Maddeler üç gruba ayrılabilir: (1) K grubu k1 ila k5 öğelerini; (2) L Grubu l1 ila l5 öğelerini; (3) Grup M ise m1 ile m5 öğelerini içerir. Destek eşiğinin% 20 olduğu varsayıldığında, aynı gruptaki öğeleri içeren öğe kümeleri sıktır, ancak farklı gruplardan öğeleri içeren öğe gruplar nadirdir. Toplam sık öğe kümesi sayısı $3 \times (2^5 - 1) = 93$'tür. Ancak, verilerde yalnızca dört sık kapalı öğe var: ({k3, k4}, {k1, k2, k3, k4, k5}, {l1, l2, l3, l4, l5} ve {m1, m2, m3, m4, m5}). Tüm sık öge kümeleri yerine yalnızca kapalı sık öğe kümelerini analistlerde kullanmak genellikle yeterlidir.



Şekil 2.7: Sık, kapalı ve maksimal öğe kümesi arasındaki ilişki [1]

Son olarak, tüm maksimum sıklıklı kümeler kapalıdır, çünkü maksimum sıklıkta olan öge kümelerinin hiçbiri, en yakın üst öğe kümeleriyle aynı destek sayısına sahip olamaz. Sık, maksimum sık ve kapalı sık öğe kümeleri arasındaki ilişkiler, Şekil 2.7'da gösterilmektedir.

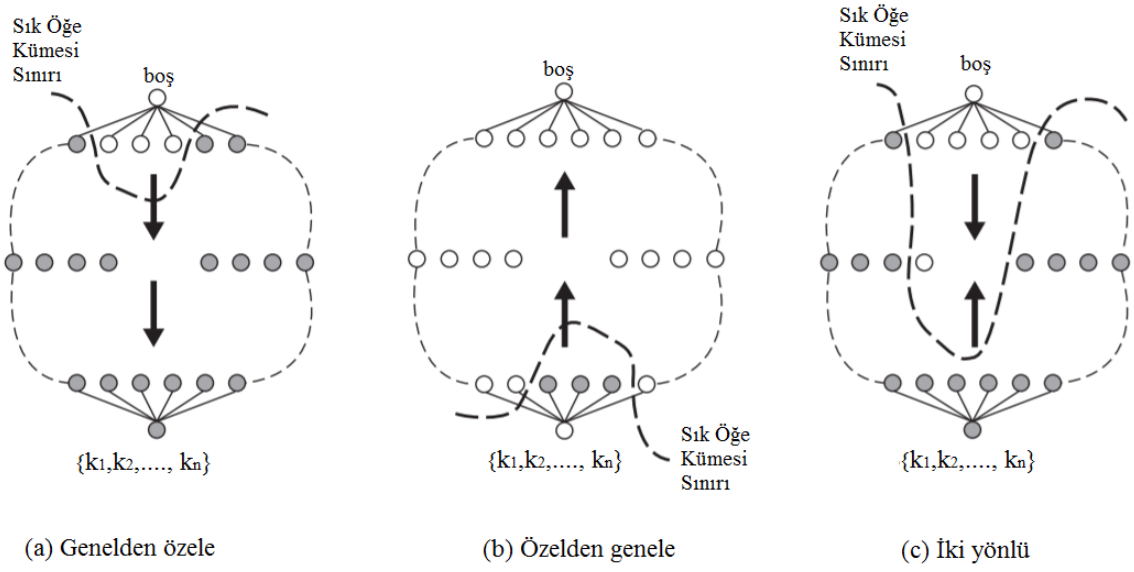
Apriori, sık ürün seti üretiminin birleştirici patlamasını başarıyla ele alan en eski algoritmalardan biridir. Üstel arama alanını budamak için Apriori prensibini uygulayarak bunu başarır. Önemli performans iyileştirmesine rağmen, algoritma işlem veri kümesi üzerinde birkaç geçiş yapılmasını gerektirdiğinden, hala önemli miktarda I / O yükü doğurmaktadır. Ek olarak, Apriori algoritmasının performansı, artan işlem genişliğinden dolayı yoğun veri kümeleri için önemli ölçüde düşebilir. Bu sınırlamaların üstesinden gelmek ve Apriori algoritmasının verimliliğini artırmak için çeşitli alternatif yöntemler geliştirilmiştir.

Bir algoritma tarafından kullanılan arama stratejisi, kafes yapısının, sık öğe kümesi oluşturma sürecinde nasıl geçtiğini belirtir. Bazı arama stratejileri, kafesteki sık öğe setlerinin yapılandırmasına bağlı olarak diğerlerinden daha iyidir.

- Genelden Özele ve Özelden genele: Apriori algoritması, aday k-öğe kümesi elde etmek için sık $(k-1)$ – öğe kümesinin birleştirildiği genel-spesifik bir arama stratejisi kullanır. Bu genel-spesifik arama stratejisi, sık bir öğe setinin maksimum uzunluğunun çok uzun

olmaması şartıyla etkilidir. Bu stratejiyle en iyi şekilde çalışan sık öğe kümelerinin yapılandırması, karanlık düğümlerin sık olmayan öğe kümelerini temsil ettiği Şekil 2.8 (a) 'da gösterilmiştir. Alternatif olarak, spesifik bir genel arama stratejisi, daha genel sık öğe kümelerini bulmadan önce, ilk önce daha spesifik sık öğe kümelerini arar. Bu strateji, Şekil 2.8 (b) 'de gösterildiği gibi, sık öğe kümesi sınırının kafesin dibine yakın olduğu yoğun işlemlerde maksimum sık öğe kümelerini keşfetmek için kullanışlıdır. Apriori prensibi, maksimum sıklıkta bulunan öge kümelerinin tüm alt kümelerini budamak için uygulanabilir. Spesifik olarak, eğer bir aday k-öğe kümesi azami sıklıkta ise, k - l büyüklüğündeki alt kümelerini incelememiz gerekmez. Bununla birlikte, eğer aday k-öğe kümesi nadirse, bir sonraki yinelemede tüm k - 1 alt kümelerini kontrol etmemiz gerekir. Diğer bir yaklaşım, hem genelden özel hem de özelden genele arama stratejilerini birleştirmektir.

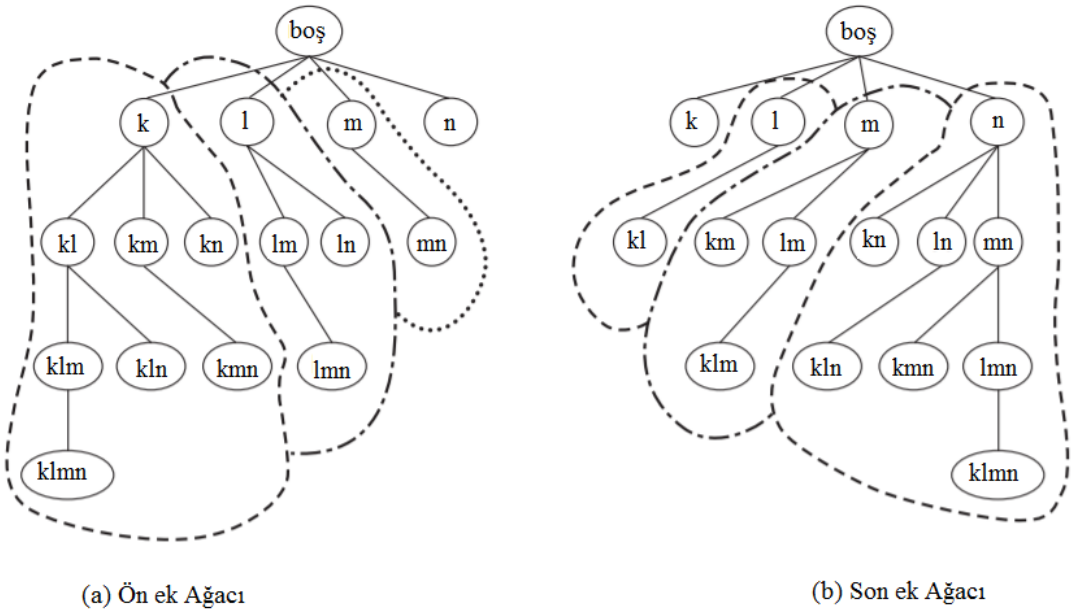
Bu iki yönlü yaklaşım, aday öğe kümelerini depolamak için daha fazla alan gerektirir, ancak Şekil 2.8 (c) 'de gösterilen yapılandırma göz önüne alındığında sık öğe kümesi sınırının hızlı bir şekilde tanımlanmasına yardımcı olabilir.



Şekil 2.8: (a) Genelden özele, (b) Özelden genele, (c) İki yönlü

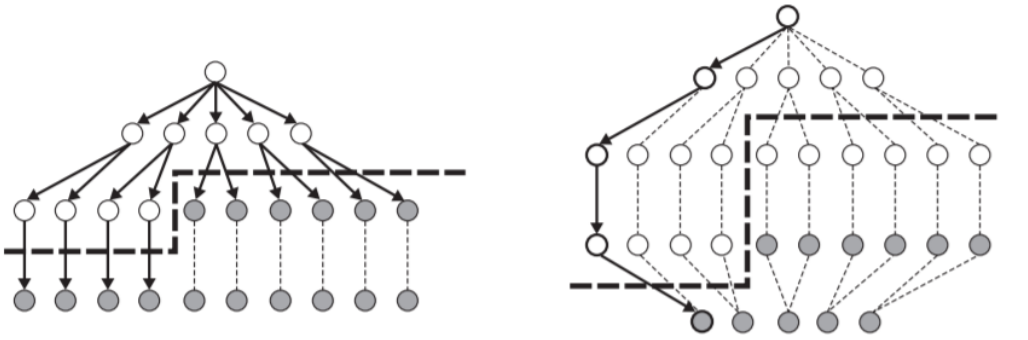
- Eşdeğerlik Sınıfları: Geçişi öngörmenin bir başka yolu, kafesin ilk önce ayrık düğüm gruplarına (veya denklik sınıflarına) bölünmesidir. Sık bir öğe kümesi üretme algoritması, başka bir denklik sınıfına geçmeden önce, belirli bir denklik sınıfındaki sık öğe kümelerini arar. Örnek olarak, Apriori algoritmasında kullanılan seviye stratejisinin

kafesin öğe kümesi boyutlarına göre bölünmesi olarak düşünülebilir; yani, algoritma, daha büyük boyutlu öğe kümelerine geçmeden önce, sık tüm 1 boyutlu öğe kümelerini keşfeder. Eşdeğerlik sınıfları ayrıca, öğe setinin ön ekine veya sonek etiketlerine göre tanımlanabilir. Bu durumda, iki öğe kümesi ortak bir ön ek veya k uzunluk sonekini paylaşırlarsa aynı denklik sınıfına aittir. Önek tabanlı yaklaşımda, algoritma, önek l, m vb. ile başlayanları aramadan önce, ön ek k ile başlayan sık öğe setlerini arayabilir. Hem önek tabanlı hem de sonek bazlı eşdeğerlik sınıfları, Şekil 2.9'da gösterilen ağaç benzeri yapı kullanılarak gösterilir.



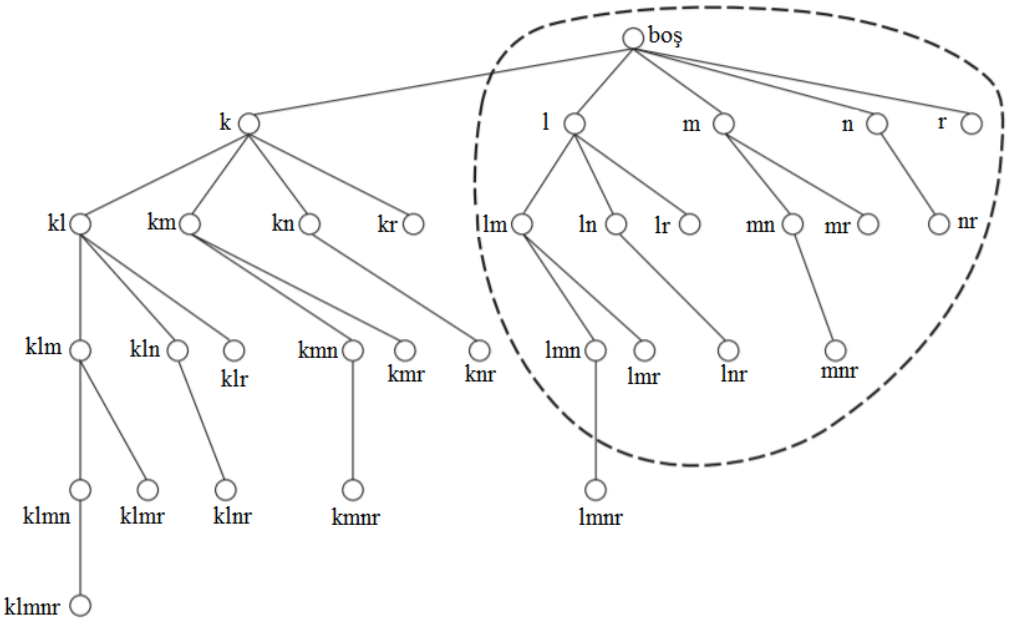
Şekil 2.9: Öğe kümelerinin önek ve sonek etiketlerine dayanan denklik sınıfları [1]

- Yayılım öncelikli arama (Breadth-First) ve derinlik öncelikli arama (Depth-First) : Apriori algoritması, kafesi, Şekil 2.9 (a) 'da gösterildiği gibi, yayılım öncelikli arama yöntemi ile dolaşır. İlk önce, sık kullanılan 1 boyutlu öğe kümelerini, ardından sık kullanılan 2 boyutlu öğe kümelerini ve daha sonra, sık yeni bir öğe seti üretilinceye kadar devam eder. Öğe kümesi kafesi Şekil 2.10(b) ve Şekil 2.11 'de gösterildiği gibi derinlik öncelikli arama yöntemi ile de dolaşabilir. Algoritma, örneğin Şekil 2.11'deki k düğümünden başlayabilir ve sık olup olmadığını belirlemek için desteğini sayabilir. Öyleyse, algoritma sonraki düğüm seviyelerini, yani kl, klm ve benzerlerini, nadiren bir düğüme erişilinceye kadar klmn gibi kademeli olarak genişletir. Sonra başka bir öğeye geri döner, örneğin, klmr ve oradan aramaya devam eder.



Şekil 2.10: Yayılım öncelikli arama ve derinlik öncelikli arama geçişleri [1]

Derinlik öncelikli yaklaşımı, çoğunlukla maksimum sık öğe setlerini bulmak için tasarlanmış algoritmalar tarafından kullanılır. Bu yaklaşım, sık öğe kümesi sınırının yayılım öncelikli arama yaklaşımı kullanmaktan daha hızlı algılanmasını sağlar. Maksimum sıklıkta bir öğe kümesi bulunduğunda, alt kümelerinde önemli budama yapılabilir. Örneğin, Şekil 2.11'de gösterilen lmn düğümü maksimum sıklıkta ise, algoritma ln, lr, m, n ve r'de yer alan alt ağaçları ziyaret etmek zorunda değildir, çünkü maksimum sıklıkta öğe kümesi içermezler. Bununla birlikte, eğer klm maksimum sıklıkta ise, sadece km ve lm gibi düğümler maksimum sıklıkta değildir (ancak km ve lm'nin alt sınıfları hala maksimum sıklıkta öğe kümeleri içerebilir). Derinlik öncelikli yaklaşımı ayrıca, öğe kümesi desteğine dayanan farklı bir budama türüne izin verir. Örneğin, {k, l, m} desteğinin {k, l} desteğiyle aynı olduğunu varsayalım. kln ve klr köklü alt ağaçlardan herhangi bir maksimum sıklıkta aday bulunmamasını garanti ettikleri için atlanabilir.



Şekil 2.11: Derinlik öncelikli yaklaşımı kullanarak aday öğe seti oluşturma

Bir işlem verisi kümesini temsil etmenin birçok yolu vardır. Temsil seçimi, aday öğe setlerinin desteğini hesaplarken oluşan I/ O maliyetlerini etkileyebilir. Tablo 2.5 ve Tablo 2.6' da, pazar sepeti işlemlerini temsil etmenin iki farklı yolunu göstermektedir. Tablo 2.5'deki gösterime Apriori de dahil olmak üzere birçok birliktelik kuralı madenciliği algoritması tarafından kabul edilen yatay veri düzeni denir.

Başka bir olasılık, her bir öğeyle ilişkili işlem tanımlayıcılarının listesini (TID) saklanmasıdır. Böyle bir temsil dikey veri düzeni olarak bilinir. Her aday öğe kümesi için destek, alt küme öğelerinin TID listelerinin kesişmesiyle sağlanır. TID listelerinin uzunluğu, daha büyük boyuttaki ürün gruplarına doğru ilerledikçe küçülür.Tablo 2.6'da ise dikey veri düzeni görülmektedir.

Tablo 2.5: Yatay Veri Düzeni

Yatay Veri Düzeni	
İşlem Numarası Listesi (TID)	**Öğeler**
1	k,l,r
2	l,m,n
3	m,r
4	k,m,n
5	k,l,m,n
6	k,r
7	k,l
8	k,l,m
9	k,m,n
10	l

Tablo 2.6: Dikey Veri Düzeni

Dikey Veri Düzeni				
k	***l***	***m***	***n***	***r***
1	1	2	2	1
4	2	3	4	3
5	5	4	5	6
6	7	8	9	
7	8	9		
8	10			
9				

Bununla birlikte, bu yaklaşımla ilgili sorun, başlangıçtaki TID listelerinin ana belleğe sığmayacak kadar büyük olabileceği ve böylece TID listelerini sıkıştırmak için daha karmaşık teknikler gerektirmesidir.

2.3.5. Minimum destek ve güven değeri

İlişkilendirme kurallarının veri tabanlarından çıkarılması genellikle iki aşamada gerçekleşir:
1. Sık öğe kümesi bulma: Bu aşamada, önceden belirlenmiş bir minimum destek değeri sağlayan öğe grupları bulunur. Birliktelik kurallarının performansını belirleyen adım budur. Bir k-öğe kümesi, 2^k - 1 sayıda boş olmayan alt kümeye sahiptir. Bu boş olmayan alt kümelerin tümü potansiyel sık öğe setleri olabilir. K değerinin büyük olduğu veri tabanlarında, bu işlemi hesaplamak ve bilgisayarlarda bile bellekte tutmak zordur. Bunun için bazı algoritmalar geliştirilmiştir. Bunlardan en yaygın olanı olan Apriori'dir.

2. Sık öğe kümelerinden güçlü birleştirme kurallarının oluşturulması: Asgari güven eşiği oluşturulduktan sonra, sık öğe kümelerinin oluşturulduğu adımdır. Sık ürün setleri tanımlandıktan sonra birleşme kurallarını oluşturmak zor bir adım değildir. Apriori özelliğine göre, her bir yinelemede minimum destek değeri sağlamayan k-öğe kümesi ortadan kalkar ve (k + 1) öğe seti üretmek için sadece eşik destek değerini sağlayan öğe setleri kullanılır.
Bu işlem, algoritma, artık sık öğe setini bulamadığı sürece devam eder. Bu işlem sayesinde, sık öğe setlerini belirlerken taranması gereken olası öğe setlerinin sayısı büyük ölçüde azaltılır. Tüm olası kurallar kümesinden ilgi kurallarını seçmek için, çeşitli önem ve ilgi ölçütleri üzerindeki çoklu kısıtlamalar kullanılabilir. En sık kullanılan kısıtlamalar, destek ve güven üzerindeki asgari eşiklerdir.

2.3.6. Güçlü birliktelik kuralları (association rules)

Birliktelik kuralı madenciliği, veri tabanındaki öğe kümeleri arasında kalıplar, birlikler ve korelasyonlar bulma sürecidir. Oluşturulan ilişkilendirme kurallarının bir öncülü ve sonucu vardır. İlişkilendirme kuralı, X & Y $\Rightarrow$ Z [destek, güven] , biçimindeki bir kalıptır. Buradaki X,Y,Z veri kümesindeki öğelerdir. Kuralın sol tarafı X & Y kuralın öncüsü ve sağ tarafı Z kuralın sonucu olarak adlandırılır. Bu, X ve Y verildiğinde Z ile bir ilişki olduğu anlamına gelir. Veri kümesi içinde, güven ve destek, her kural için kesinliği veya yararı belirlemeye yönelik iki ölçüttür. Destek, veri kümesindeki bir dizi öğenin hem öncülü hem de kuralın sonucunu içermesi olasılığıdır, P (X $\cup$ Y $\cup$ Z). Güven, öncülü içeren bir dizi öğenin de sonucu içerme olasılığıdır, P (Z| X $\cup$ Y). Tipik olarak, hem minimum destek eşiğini hem de kullanıcı tarafından belirlenen minimum güven eşiğini karşılarsa, bir ilişkilendirme kuralı güçlü olarak adlandırılır. [6]

Örneğin, Bir spor malzemeleri mağazasının, tek bir satın alma sırasında birlikte satın alınan ürünler arasında herhangi bir ilişkilendirme kuralı olup olmadığını belirlemek istediğini varsayalım.

Tablo 2.7: Spor malzemeleri işlemleri [6]

İşlem No	**Öğeler**
1	forma, dişlik, pantolon
2	forma, pantolon
3	krampon, forma
4	şapka, dişlik, tenis topu

Bulabileceğiniz bir kural forma $\Rightarrow$ pantolon. Bu kural için destek, bir işlemin forma ve pantolonu birlikte içermesi olasılığıdır. Tablo 2.7'ye göre bu kural 4 işlemde 2 kere gerçekleşmiştir yani destek 2/4 veya %50 'dir.
Güven, forma içeren bir işlemin pantolonu da içerme olasılığıdır. Forma içeren 3 işlem ve forma ile pantolonu beraber içeren 2 işlem olduğundan dolayı güven 2/3 veya % 66'dır.
Bu nedenle tam şekliyle yazılmış bu kural forma $\Rightarrow$ pantolon [% 50,% 66] 'dır.

Eğer pantolon öncül olarak seçilmiş olsaydı kural pantolon $\Rightarrow$ forma [% 50,% 100] olurdu. Bu durumda güven % 100'dür çünkü pantolon içeren her işlemde aynı zamanda forma da vardır. Bu ilişkilendirme kuralları, küçük veri kümesi olması nedeniyle oluşturulması oldukça kolaydır, ancak veri kümesi büyüdükçe zorlaşır.

Büyük veri kümelerindeki veri madenciliğini daha iyi anlamak için bazı temel terim ve kavramların anlaşılması gerekir. K adet öğe içeren öğe seti, bir k-öğe set'tir. Örneğin; {A,B} kümesi 2 öğeli bir kümedir. Bir öğe setinin ortaya çıkma sıklığı, sadece öğe seti içeren işlemlerin sayısıdır. Bazen, öğe setinin frekansı, destek sayısı veya öğe setinin sayısı olarak da adlandırılır. Minimum destek sayısı, öğe setinin minimum desteği sağlaması için gereken işlem sayısı olarak tanımlanır. Minimum destek sayısı, veri kümesindeki toplam işlem sayısının ve kullanıcı tanımlı minimum desteğin ürününe eşittir.

Asgari desteği sağlayan herhangi bir öğe kümesi, sık öğe kümesi olarak kabul edilir ve k öğe öğe kümesi genel olarak L_k ile gösterilir.

Tablo 2.7'deki örneğe geri dönersek aşağıdaki değerleri alabiliriz. Öğe kümesi {forma} oluşum sıklığı 3'e eşittir, çünkü veri tabanındaki işlemlerin 3'ünde meydana gelir. Minimum destek, *toplam işlem sayısı * destek* işlemine eşittir veya yukarıdaki durumda 4 x 50 = 2 olacaktır. Bunun nedeni, daha önce % 50'lik asgari bir destek eşiği belirlemiş olmamızdır. Bu nedenle, 1-öğe seti, {forma}, asgari desteği sağlar ve bu nedenle sık bir öğe seti olarak kabul edilir ve L_1 'de bulunur.

Büyük veri kümelerindeki birliktelik kural madenciliği iki aşamaya bölünmüştür. İlk aşama, önceden belirlenmiş asgari destek sayısı kadar sık gerçekleşen tüm öğe kümelerini bulmaktır. Bu adım sık öğe setlerinin L_1 'den L_k 'ya kadar k listelerini üretecektir. İkinci adım, sık öğe kümelerinden güçlü birleştirme kuralları oluşturmaktır. Hem asgari destek hem de asgari güveni sağlaması halinde bir birliktelik kuralı güçlü olarak kabul edilir.

Güçlü birliktelik kuralı $\Rightarrow$ destek(R) >= minimum destek ve güven (R) >= minimum güven

2.4. Sık Geçen Nesne Kümeleri Madenciliği

2.4.1. Apriori algoritması

Apriori algoritması, bir aday veri setini kullanarak bir veri setinden sık öğe setlerini bulmak için kullanılan temel bir algoritmadır. Apriori, (k + 1) öğe setini belirlemek için k-öğe setini kullanıldığından dolayı seviye bazında arama olarak bilinen yinelemeli bir yaklaşım kullanır. Arama, L_1 ile gösterilen sık 1-öğe kümesi ile başlar. L_1 daha sonra sık 2-öğe seti L_2 yi bulmak için kullanılır. L_2 daha sonra L_3 'ü bulmak için kullanılır. Bu, Şekil 2.12'de görüldüğü üzere daha sık k-öğe seti bulunamayana kadar devam eder. [6]

Apriori(T, ∈)

$L_1 \leftarrow$ *{büyük 1 – öğekümeleri}*

$k \leftarrow 2$

while $L_{k-1} \neq \emptyset$

$C_k \leftarrow \{c = a \cup \{b\} \mid a \in L_{k-1} \wedge b \notin a, \{ s \subseteq c \mid\mid s\mid = k-1 \} \subseteq L_{k-1}\}$

for *işlem* $t \in T$

$D_t \leftarrow \{c \in C_k \mid c \subseteq t\}$

for *adaylar* $c \in D_t$

sayaç[c] $\leftarrow$ *sayaç[c] + 1*

$L_k \leftarrow \{c \in C_k \mid sayaç[c] \geq \in \}$

$k \leftarrow k + 1$

return $\bigcup_k L_k$

Şekil 2.12: Apriori algoritma formulü [1]

Seviyelendirilmiş bir neslin verimliliğini artırmak için Apriori algoritmasını apriori özelliğini kullanır. Apriori özelliği, sık bir öğe kümesinin tüm boş olmayan alt kümelerinin de sık bir öğe kümesi olduğunu belirtir.

Öyleyse, {A, B} sık bir öğe kümesi ise, {A} ve {B} alt kümeleri de sık öğe kümesidir. Seviye arama, bu apriori özelliğini, bir seviyeden diğerine geçerken kullanır. *I* bir öğe seti asgari desteği sağlamazsa, o zaman *I* sık bir öğe seti olarak kabul edilmez. *A* öğesi, *I* öğe kümesine eklenirse, yeni öğe kümesi $I \cup A$, I orijinal öğe kümesinden daha sık meydana gelemez. Bir öğe kümesi sık bir öğe kümesi olarak değerlendirilmezse, o öğe grubunun tüm üst kümeleri de aynı sınamada başarısız olur. Apriori algoritması, aday listesindeki öğe sayısını azaltmak ve dolayısıyla arama süresini optimize etmek için bu özelliği kullanır. Apriori algoritması L_{k-1} den L_k i bulmak için Katılma adımı (Join Step) ve Budama adımı (Prune Step) dan oluşan iki aşamalı bir süreç kullanır.

İlk adım Join Steptir ve L_{k-1}'den C_k olarak adlandırılan bir takım aday k-öğe seti oluşturmaktan sorumludur. Bunu, L_{k-1} 'i kendisiyle birleştirerek yapar. Apriori, bir işlemdeki veya öğe kümesindeki öğelerin sözlükbilim sırasına göre sıralandığını varsayar.

L_{k-1} ila L_{k-1} 'in birleştirilmesi, yalnızca birbirleriyle ortak olan ilk (k-2) öğelere sahip olan öğeler arasında gerçekleştirilir. Farz edelim ki I_1 ve I_2 öğeleri L_{k-1}'in üyeleridir. Eğer (($I_1[1] = I_2[1]$ ve $I_1[2] = I_2[2]$ ve… ve $I_1[k-2] = I_2[k-2]$ ve $I_1[k-1] < I_2[k-1]$) ise birbirleriyle birleştirilirler. Burada $I_1[1]$ ve I_1 kümesindeki ilk öğe, $I_1[k-1]$ son öğedir ve I_2 için böyle devam eder.

İkinci adım Prune Step'tir ve C_k 'yı L_k 'ya dönüştürür. Aday C_k listesi sık kullanılan k-öğe setlerinin tümünü içerir, fakat aynı zamanda minimum destek sayımını karşılamayan k-öğe setlerini de içerir. Veritabanının taranması, asgari desteği sağlayıp sağlamadığını belirlemek için her aday k-öğe setinin ortaya çıkma sıklığını belirleyecektir. C_k büyüdükçe bu çok maliyetli olacaktır. C_k boyutunu azaltmak için apriori özelliği kullanılır. Apriori özelliği, sık olmayan bir (k-1) öğe kümesinin, sık bir k-öğe kümesinin alt kümesi olamayacağını belirtir. [1]

2.4.1.1. Apriori özelliği

Bu algoritmanın iki önemli özelliği vardır:

1. Sıralı bir paternin ilerlemesine eşlik eden bir değerlendirme kriterinin monotonik azalmasını ifade eder. Tüm sık sıralı düzenleri etkili bir şekilde keşfetmek için etkinleştirilir.
2. Apriori özelliği, sıralı modellerin değerlendirme kriterlerinin değerlerinin sıralı alt modellerinin değerlerinden küçük veya eşit olduğunu gösteren özelliktir. [11]

2.4.1.2. Apriori işleyişi

Apriori algoritmasında anahtar kavram destek ölçüsünün anti-monotonikliğidir. Aşağıdaki durumları varsayar; [7]

1. Sık öge setinin tüm alt kümeleri sık olmalıdır.
2. Benzer şekilde, nadir bulunan bir öğe için, tüm üst kümesi de nadir olmalıdır.

Örneğin, sürece başlamadan önce, destek eşiğini % 50'ye ayarlayalım, yani yalnızca desteğin % 50'den fazla olduğu maddeler önemlidir.

Adım 1: Tüm işlemlerde oluşan tüm öğelerin sıklık tablosu oluşturulmalıdır.

Adım 2: Desteğin eşik desteğine eşit veya ondan daha büyük olduğu sadece bu unsurların önemli olduğunu bilinir. Burada, destek eşiği % 50'dir, bu nedenle yalnızca üçten fazla işlemde ortaya çıkan öğeler önemlidir ve bu öğeler Soğan (S), Patates (P), Ekmek (E) ve Kola (K) 'dir.

Tablo 2.8: Basit Apriori Algoritması Örneği 1 [7]

Öğe	*İşlem Sayısı*
Soğan(S)	4
Patates (P)	5
Ekmek (E)	4
Kola (K)	4

Adım 3: Bir sonraki adım, siparişin önemli olmadığını, yani AB'nin BA ile aynı olduğunu göz önünde bulundurarak, önemli kalemlerin tüm olası çiftlerini yapmaktır. Bunu yapmak için, ilk öğe alınır ve SP, SE, SK gibi diğerleri ile eşleştirilir. Benzer şekilde, ikinci maddeyi göz önünde bulundurulur ve önceki maddelerle, yani PE, PK ile eşleştirilir. PS'nun (SP ile aynı) zaten olduğu için sadece önceki maddeleri düşünülür. Yani, örneğimizdeki tüm çiftler SP, SE, SK, PE, PK, EK'dir.

Adım 4: Her işlemdeki her bir çiftin oluşumu sayılır.

Tablo 2.9: Basit Apriori Algoritması Örneği 2 [7]

Öğe	*İşlem Sayısı*
SP	4
SE	3
SK	2
PE	4
PK	3
EK	2

Adım 5: Yine, sadece destek eşiğini geçen bu ürünler ve SP, SE, PE ve PK olanlar önemlidir.

Adım 6: Şimdi birlikte satın alınan üç öğeden oluşan bir dizi aramak istediğimizi varsayalım. 5. adımda bulunan öğeler kullanılır ve 3 maddeden oluşan bir küme yaratılır.
3 maddeden oluşan bir küme oluşturmak için, kendine katılmak adı verilen başka bir kural gerekir. Öyle ki SP, SE, PE ve PK madde çiftlerinden aynı ilk harfle iki çift aranır .

SP ve SE, bu SPE'yi verir.
PE ve PK, bu PEK verir.

Daha sonra, bu iki öğe için frekans bulunur.

Tablo 2.10: Basit Apriori Algoritması Örneği 3 [7]

Öğe	*İşlem Sayısı*
SPE	4
PEK	3

Eşik kuralını tekrar uygulayarak SEP'nin tek önemli öğeler olduğu görülür.

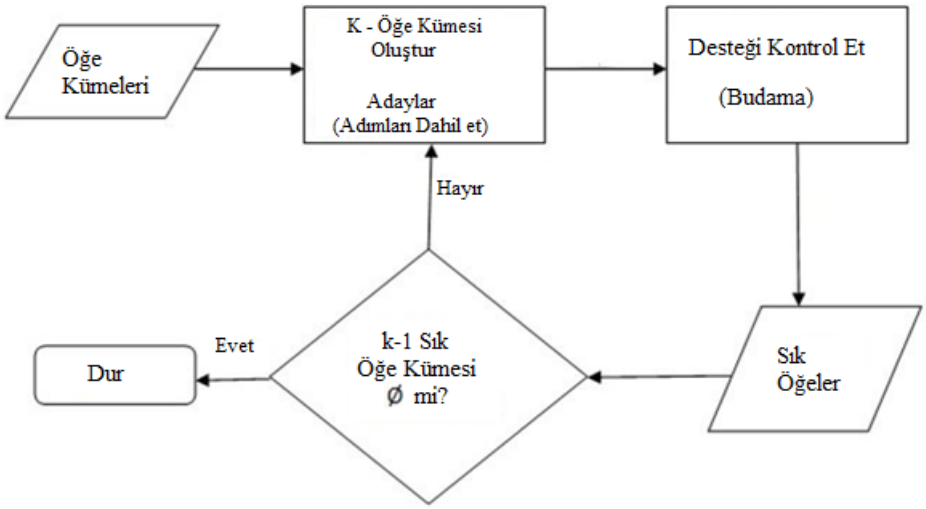
Bu nedenle, en sık satın alınan 3 ürün kümesi SPE dir.

Örnek oldukça basit ve sık öğelerin 3 öğede durdurulduğunu fakat pratikte düzinelerce madde bulunduğunu ve bu işlemin birçok öğeye kadar devam edebileceğini gösterir.

ABC, ABD, ACD, ACE ve BCD olarak 3 uzunluklu öğe setlerimiz olduğunu ve 4 uzunluklu öğe setlerini oluşturmak istediğimizi varsayalım. Bunun için ilk iki alfabenin ortak olduğu kümelere bakılır, yani

ABC ve **AB**D **AB**CD'i verir.
ACD ve **AC**E, **AC**DE'yi verir.

Tüm algoritma iki aşamaya ayrılabilir:
Adım 1 : Bir veritabanında k öğeleriyle birlikte tüm sık kümeleri bulmak için minimum destek uygulanır.
Adım 2 : k + 1 öğeli sık kümeleri sık k-öğe kümelerinin yardımıyla bulmak için kendi kendine katılma kuralı kullanılır. Kendi kendine katılma kuralını uygulanamayana kadar bu işlem (k=1) tekrarlanır. Bu sürecin işleyişi Şekil 2.13'de görünmektedir.



Şekil 2.13: Apriori Algoritmasının Genel Süreci [7]

2.4.2. Eclat algoritması

Eclat, bir işlem veritabanında (sık ögeler) sıkça oluşan öğeler kümelerini (öğe grubu) keşfetmeye yönelik bir algoritmadır. Zaki (2001) tarafından önerilmiştir. Apriori gibi algoritmaların aksine, Eclat, yayılım öncelikli arama (breadth first search) yerine sık kullanılan öğeleri bulmak için derin öncelikli arama (depth first search) kullanır. [3]

Sık bir öğe kümesi, işlem veritabanında minumum destek (minsup) işlemlerinde görünen bir öğe kümesidir, burada minumum destek (minsup), kullanıcı tarafından verilen bir parametredir. [8]

Eclat algoritması ilginçtir çünkü derin öncelikli aramayı kullanır. Veriler dikey biçimde hazırlanır. Apriori ve FP-Growth algortimaları ise yatay olarak verileri biçimlendirir. [9]

Tablo 2.11: Dikey biçimdeki veriseti örneği(Eclat) [3]

İşlem Numarası Listesi (TID)	*Öğeler*
Ekmek	1,4,5,7,8,9
Tereyağı	1,2,3,4,6,8,9
Süt	3,5,6,7,8,9
Soda	2,4
Reçel	1,8

Tablo 2.12: Yatay biçimdeki veriseti örneği(Apriori) [3]

İşlem Numarası Listesi (TID)	*Öğeler*
1	Ekmek, Tereyağı, Reçel
2	Tereyağı, Soda
3	Tereyağı, Süt
4	Ekmek, Tereyağı, Soda
5	Ekmek, Süt
6	Tereyağı, Süt
7	Ekmek, Süt
8	Ekmek, Tereyağı, Süt, Reçel
9	Ekmek, Tereyağı, Süt

Eclat algortiması adımları :

- Her öğe için TID listesini alınır. (Veriseti taraması)
- {Ekmek} 'in TID listesi, tam olarak {Ekmek} içeren işlemlerin listesidir.
- {Ekmek} ‘in diğer öğeler ile 2 li öğe setlerine bakılır.
- {Ekmek} ‘in diğer öğelerle olan 3 ve fazlası öğe setlerine bakılır.
- Bu işlemler tüm öğeler için yapılır.

Tablo 2.13: İşlemin 1.adımı

Sık 1-öğekümesi	*Öğe Kümesi*	*İşlem Numarası Listesi (TID)*
Minimum Destek (min_sup) =2	Ekmek	1,4,5,7,8,9
	Tereyağı	1,2,3,4,6,8,9
	Süt	3,5,6,7,8,9
	Soda	2,4
	Reçel	1,8

Tablo 2.14: İşlemin 2. adımı

Sık 2-öğekümesi	*Öğe Kümesi*	*İşlem Numarası Listesi (TID)*
	{Ekmek, Tereyağı}	1,4,8,9
	{Ekmek, Süt}	5,7,8,9
	{Ekmek, Soda}	4
	{Ekmek, Reçel}	1,8
	{Tereyağı, Süt}	3,6,8,9
	{Tereyağı, Soda}	2,4
	{Tereyağı, Reçel}	1,8
	{Süt, Reçel}	8

Tablo 2.15: İşlemin 3.adımı

Sık 3-öğekümesi	*Öğe Kümesi*	*İşlem Numarası Listesi (TID)*
	{Ekmek, Tereyağı, Süt}	8,9
	{Ekmek, Tereyağı, Reçel}	1,8

Bu işlem sık öğeye veya hiçbir aday öğeye rastlanmayana kadar k her seferinde 1 artırılarak tekrar eder. k > = 1 için (k + 1) öğe setinin desteğini bulmak için veritabanını taramaya gerek yoktur. Derinlik öncelikli arama kullanıldığından dolayı bellek gereksinimlerini azaltır bu sebeple genellikle (önemli ölçüde) apriori'den daha hızlıdır.

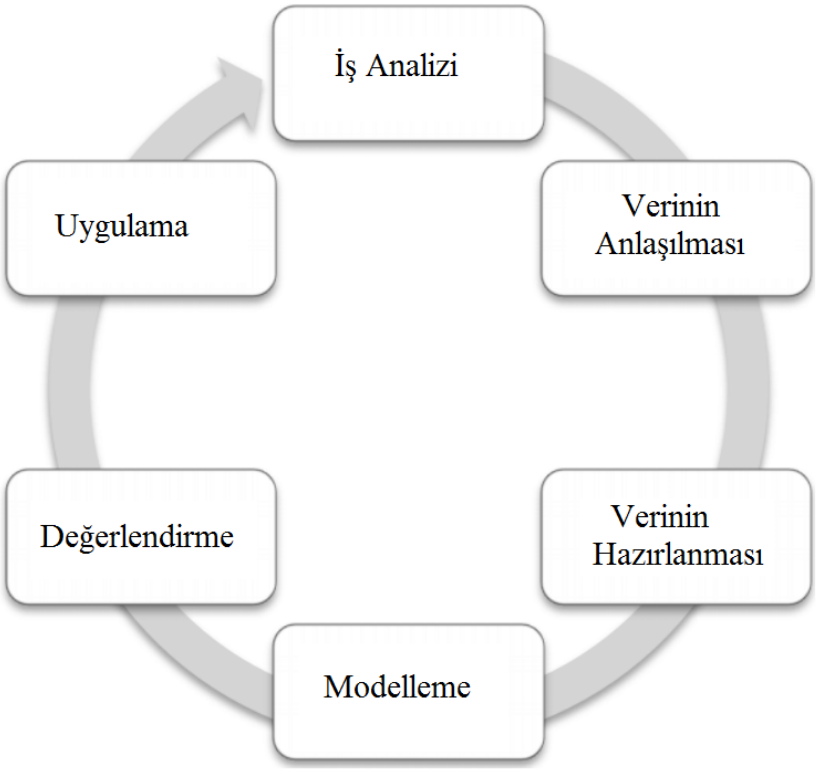
3. SİSTEMİN TASARIMI

Veri madenciliği, makine öğrenmesi, yapay zeka (AI) ve istatistik gibi çeşitli veri madenciliği teknikleri kullanılarak, veri tabanlarında veya veri ambarında depolanan büyük miktarda veriyi analiz ederek gizli değerli bilgileri keşfetme süreci olarak tanımlanır.

Çeşitli endüstrilerdeki birçok kuruluş, iş verimliliğini artırmak için üretim, pazarlama, kimyasal, yazılım, havacılık vb. dahil olmak üzere veri madenciliğinden yararlanır. Bu nedenle, standart bir veri madenciliği sürecine duyulan ihtiyaç çarpıcı biçimde artmıştır.

Sonuç olarak, 1990'da veri madenciliği Cross Industry Standard Process for Data Mining(CRISP-DM) için bir çok atölye çalışmasından ve ilk olarak 300'den fazla kuruluşun katkısından sonra sektörler arası standart bir süreç yayınlanmıştır.[10]

CRISP-DM, bir veri madenciliği projesinin yaşam döngüsünü Şekil 3.1'de görüldüğü üzere altı aşamaya ayırır.



Şekil 3.1: Veri madenciliği yaşam döngüsü [10]

3.1. İş Analizi

Bu aşamada; proje amaçlarını, gereksinimlerini iş perspektifinden anlamaya, daha sonra bu bilgiyi veri madenciliği problemi tanımına ve hedeflere ulaşmak için tasarlanmış bir ön proje planına dönüştürmeye odaklanır.

3.2. Verinin Anlaşılması

Veri anlama aşaması, ilk veri toplama ile başlar ve verilere aşina olma, veri kalitesi problemlerini belirleme, verilere ilişkin ilk bilgileri keşfetme veya gizli bilgiler için hipotezler oluşturmak için ilginç altkümeleri saptamak için faaliyetlerle devam eder.

İş analizi ile verinin anlaşılması arasında yakın bir bağlantı vardır. Veri madenciliği sorununun ve proje planının formülasyonu, en azından mevcut verilerin anlaşılmasını gerektirir. [10]

3.3. Verinin Hazırlanması

Veri hazırlama aşaması, başlangıçtaki ham verilerden nihai veri setini (modelleme araç(lar)ının besleneceği veriler) oluşturmak için tüm faaliyetleri kapsar. Veri hazırlama görevlerinin birden fazla defa yapılması ve önceden belirlenmiş bir sırada yapılmaması muhtemeldir. Görevler tablo, kayıt ve özellik seçimi, veri temizliği, yeni özelliklerin oluşturulması ve modelleme araçları için verilerin dönüşümünü içerir. [10]

3.4. Modelleme

Bu aşamada çeşitli modelleme teknikleri seçilip uygulanır ve parametreleri optimum değerlere ayarlanır. Tipik olarak, aynı veri madenciliği problem türü için birkaç teknik vardır. Bazı teknikler belirli veri formatları gerektirir. Veri Hazırlama ve Modelleme arasında yakın bir bağlantı vardır. Genellikle modelleme sırasında veri problemleri fark edilirken, bir diğer taraftan yeni veri oluşturmak için fikir edinilir. [10]

3.5. Değerlendirme

Modelin son dağıtımına geçmeden önce, modeli daha kapsamlı bir şekilde değerlendirmek ve modeli oluşturmak için atılan adımları gözden geçirmek, hedeflere uygun şekile ulaştığından emin olmak önemlidir.

Anahtar amaç, yeterince düşünülmemiş bazı önemli iş konularının olup olmadığını belirlemektir. Bu aşama sonunda, veri madenciliği sonuçlarının kullanımı konusunda bir karara varılmalıdır. [10]

3.6. Uygulama

Modelin oluşturulması genellikle projenin sonu değildir. Genellikle, kazanılan bilginin kullanabilecek şekilde düzenlenmesi ve sunulması gerekecektir. Gereksinimlere bağlı olarak, dağıtım aşaması bir rapor oluşturmak kadar basit veya tekrarlanabilir bir veri madenciliği sürecinin uygulanması kadar karmaşık olabilir. Her durumda, oluşturulan modellerden gerçekten faydalanmak için hangi işlemlerin yapılması gerektiğini anlamak önemlidir. [10]

4. UYGULAMA

4.1 İş Analizi

Bu çalışmanın amacı, bir üniversiteye ait öğrencilerin ders esnasındaki internet kullanımları analiz edilerek dersteki ilgi durumlarının tespit edilmesidir. Şekil 4.1'de çalışmanın genel kapsamı ifade edilmiştir.

Çalışmada lisans gerektirmeyen açık kaynak olan Anaconda ve R Studio geliştirme ortamlarında Apriori algoritması kullanılarak yapılmıştır.

4.2. Verinin Anlaşılması ve Modellenmeye Hazırlanması

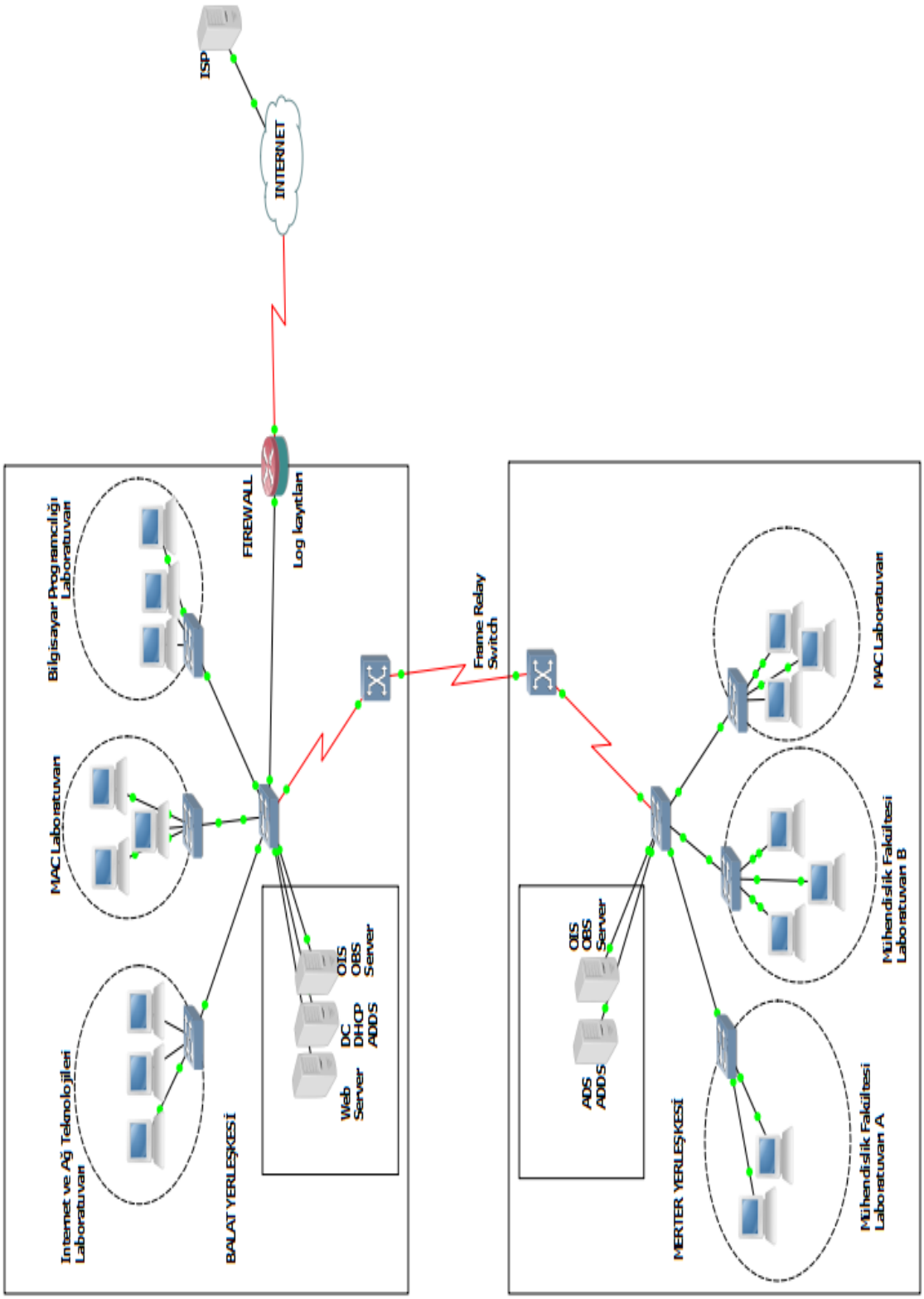
Bu aşamada veriyi elde etme, verinin kullanılacak olan algoritmaya göre modellenmesi, gereksiz ve yanlış yönlendirmelere sebebiyet verecek verilerin temizlenmesi işlemleri yapılacaktır.

Bu çalışmada, Ayvansaray Üniversitesi Plato Meslek Yüksek Okulu İnternet ve Ağ Teknolojileri bölüm laboratuvarındaki network trafiği incelenmiştir.

01.03.2018 ve 10.05.2018 tarihleri arasındaki öğrencilerin ders saatleri içerisinde (15:00 – 18:00) toplam 9 haftalık internet kullanımı kontrol edilmiştir.

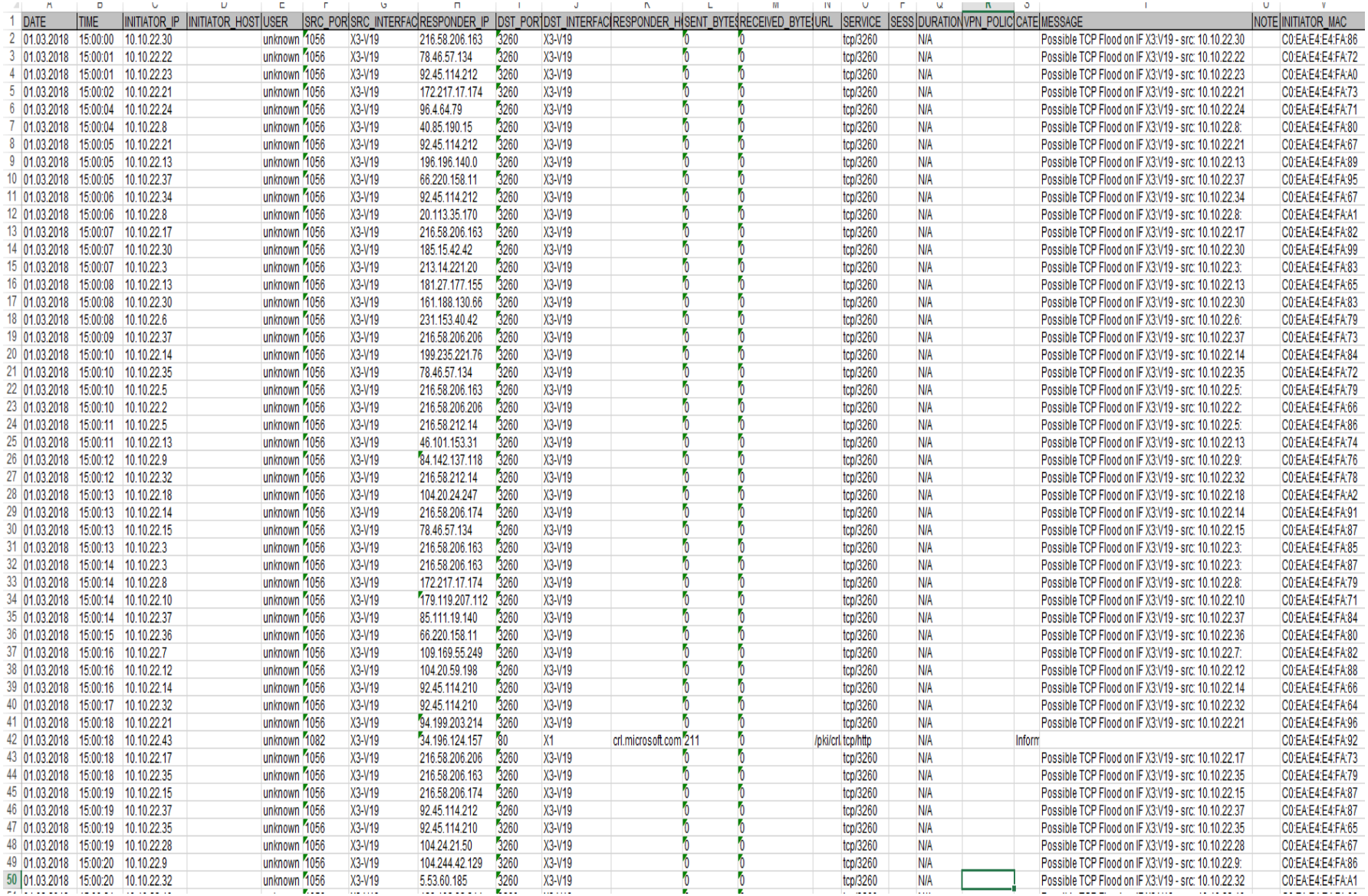
4.2.1 Verinin ağdan alınması

Çalışmamızda kullanılan verinin elde edilmesi için network trafiğinin izlenmesi gerekmektedir. Bunun için farklı yollar olmasına rağmen en düzgün ve kapsamlı bilgiyi okulun internete çıkan dış ağ bağlantı yoluna sahip güvenlik duvarı(firewall) üzerinden trafiğin kontrol edilmesiyle olacaktır. Bu sebeple Ayvansaray Üniversitesi etik kurulu onayı ile güvenlik duvarı(firewall) üzerindeki loglar alınmıştır. Verilerin ağdan alınması işlemi Şekil 4.1 de gösterildiği gibidir.



Şekil 4.1: Ağdan verinin alınması

Şekil 4.2 de gösterilen firewall log kayıtları metin dosyası sadece bir hafta içerisinde ilgili derse ait günün verilerini göstermektedir. Bunun gibi toplam dokuz haftaya ait metin dosyası bulunmaktadır. Dosyaların içerisinde sırasıyla; tarih, saat, kurum içinde kullanılan IP'ler (initiator IP), kurum içinde kullanılan host (initiator host), kullanıcı(user), kaynak port (source port), kaynak ara yüz (source interface), hedef IP (responder IP), hedef port (destination port), hedef ara yüz (destination interface), hedef host (responder host), gönderilen datanın byte karşılığı (sent bytes), alınan datanın byte karşılığı (received bytes), sertifika URL'i, servis, oturum bilgileri (session), süre (duration), vpn kuralları, kategori, mesaj , not, kurum içinde kullanılan MAC adresleri (initiator MAC), gidilen MAC adresleri (responder MAC), hedeflenen MAC adresi (target MAC), firewall action, kural(rule), dosya ID (file ID) ve durum (status) alanları yer almaktadır. Bu alanlar Şekil 4.2'de görüldüğü üzere, bazılarının değerleri güvenlik duvarı(firewall) kaynaklı boş gelmiştir.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	DATE	TIME	INITIATOR_IP	INITIATOR_HOST	USER	SRC_POR	SRC_INTERFAC	RESPONDER_IP	DST_POR	DST_INTERFAC	RESPONDER_H	SENT_BYTES	RECEIVED_BYTE	URL	SERVICE	SESS	DURATION	VPN_POLIC	CATE	MESSAGE	NOTE	INITIATOR_MAC
2	01.03.2018	15:00:00	10.10.22.30		unknown	1056	X3-V19	216.58.206.163	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.30		C0:EA:E4:E4:FA:86
3	01.03.2018	15:00:01	10.10.22.22		unknown	1056	X3-V19	78.46.57.134	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.22		C0:EA:E4:E4:FA:72
4	01.03.2018	15:00:01	10.10.22.23		unknown	1056	X3-V19	92.45.114.212	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.23		C0:EA:E4:E4:FA:A0
5	01.03.2018	15:00:02	10.10.22.21		unknown	1056	X3-V19	172.217.17.174	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.21		C0:EA:E4:E4:FA:73
6	01.03.2018	15:00:04	10.10.22.24		unknown	1056	X3-V19	96.4.64.79	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.24		C0:EA:E4:E4:FA:71
7	01.03.2018	15:00:04	10.10.22.8		unknown	1056	X3-V19	40.85.190.15	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.8:		C0:EA:E4:E4:FA:80
8	01.03.2018	15:00:05	10.10.22.21		unknown	1056	X3-V19	92.45.114.212	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.21		C0:EA:E4:E4:FA:67
9	01.03.2018	15:00:05	10.10.22.13		unknown	1056	X3-V19	196.196.140.0	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.13		C0:EA:E4:E4:FA:89
10	01.03.2018	15:00:05	10.10.22.37		unknown	1056	X3-V19	66.220.158.11	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.37		C0:EA:E4:E4:FA:95
11	01.03.2018	15:00:06	10.10.22.34		unknown	1056	X3-V19	92.45.114.212	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.34		C0:EA:E4:E4:FA:67
12	01.03.2018	15:00:06	10.10.22.8		unknown	1056	X3-V19	20.113.35.170	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.8:		C0:EA:E4:E4:FA:A1
13	01.03.2018	15:00:07	10.10.22.17		unknown	1056	X3-V19	216.58.206.163	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.17		C0:EA:E4:E4:FA:82
14	01.03.2018	15:00:07	10.10.22.30		unknown	1056	X3-V19	185.15.42.42	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.30		C0:EA:E4:E4:FA:99
15	01.03.2018	15:00:07	10.10.22.3		unknown	1056	X3-V19	213.14.221.20	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.3:		C0:EA:E4:E4:FA:83
16	01.03.2018	15:00:08	10.10.22.13		unknown	1056	X3-V19	181.27.177.155	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.13		C0:EA:E4:E4:FA:65
17	01.03.2018	15:00:08	10.10.22.30		unknown	1056	X3-V19	161.188.130.66	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.30		C0:EA:E4:E4:FA:83
18	01.03.2018	15:00:08	10.10.22.6		unknown	1056	X3-V19	231.153.40.42	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.6:		C0:EA:E4:E4:FA:79
19	01.03.2018	15:00:09	10.10.22.37		unknown	1056	X3-V19	216.58.206.206	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.37		C0:EA:E4:E4:FA:73
20	01.03.2018	15:00:10	10.10.22.14		unknown	1056	X3-V19	199.235.221.76	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.14		C0:EA:E4:E4:FA:84
21	01.03.2018	15:00:10	10.10.22.35		unknown	1056	X3-V19	78.46.57.134	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.35		C0:EA:E4:E4:FA:72
22	01.03.2018	15:00:10	10.10.22.5		unknown	1056	X3-V19	216.58.206.163	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.5:		C0:EA:E4:E4:FA:79
23	01.03.2018	15:00:10	10.10.22.2		unknown	1056	X3-V19	216.58.206.206	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.2:		C0:EA:E4:E4:FA:66
24	01.03.2018	15:00:11	10.10.22.5		unknown	1056	X3-V19	216.58.212.14	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.5:		C0:EA:E4:E4:FA:86
25	01.03.2018	15:00:11	10.10.22.13		unknown	1056	X3-V19	46.101.153.31	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.13		C0:EA:E4:E4:FA:74
26	01.03.2018	15:00:12	10.10.22.9		unknown	1056	X3-V19	64.142.137.118	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.9:		C0:EA:E4:E4:FA:76
27	01.03.2018	15:00:12	10.10.22.32		unknown	1056	X3-V19	216.58.212.14	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.32		C0:EA:E4:E4:FA:78
28	01.03.2018	15:00:13	10.10.22.18		unknown	1056	X3-V19	104.20.24.247	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.18		C0:EA:E4:E4:FA:A2
29	01.03.2018	15:00:13	10.10.22.14		unknown	1056	X3-V19	216.58.206.174	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.14		C0:EA:E4:E4:FA:91
30	01.03.2018	15:00:13	10.10.22.15		unknown	1056	X3-V19	78.46.57.134	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.15		C0:EA:E4:E4:FA:87
31	01.03.2018	15:00:13	10.10.22.3		unknown	1056	X3-V19	216.58.206.163	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.3:		C0:EA:E4:E4:FA:85
32	01.03.2018	15:00:14	10.10.22.3		unknown	1056	X3-V19	216.58.206.163	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.3:		C0:EA:E4:E4:FA:87
33	01.03.2018	15:00:14	10.10.22.8		unknown	1056	X3-V19	172.217.17.174	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.8:		C0:EA:E4:E4:FA:79
34	01.03.2018	15:00:14	10.10.22.10		unknown	1056	X3-V19	179.119.207.112	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.10		C0:EA:E4:E4:FA:71
35	01.03.2018	15:00:14	10.10.22.37		unknown	1056	X3-V19	85.111.19.140	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.37		C0:EA:E4:E4:FA:84
36	01.03.2018	15:00:15	10.10.22.36		unknown	1056	X3-V19	66.220.158.11	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.36		C0:EA:E4:E4:FA:80
37	01.03.2018	15:00:16	10.10.22.7		unknown	1056	X3-V19	109.169.55.249	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.7:		C0:EA:E4:E4:FA:82
38	01.03.2018	15:00:16	10.10.22.12		unknown	1056	X3-V19	104.20.59.198	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.12		C0:EA:E4:E4:FA:88
39	01.03.2018	15:00:16	10.10.22.14		unknown	1056	X3-V19	92.45.114.210	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.14		C0:EA:E4:E4:FA:66
40	01.03.2018	15:00:17	10.10.22.32		unknown	1056	X3-V19	92.45.114.210	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.32		C0:EA:E4:E4:FA:64
41	01.03.2018	15:00:18	10.10.22.21		unknown	1056	X3-V19	94.199.203.214	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.21		C0:EA:E4:E4:FA:96
42	01.03.2018	15:00:18	10.10.22.43		unknown	1082	X3-V19	34.196.124.157	80	X1	crl.microsoft.com	211	0	/pki/crl	tcp/http		N/A			Inform		C0:EA:E4:E4:FA:92
43	01.03.2018	15:00:18	10.10.22.17		unknown	1056	X3-V19	216.58.206.206	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.17		C0:EA:E4:E4:FA:73
44	01.03.2018	15:00:18	10.10.22.35		unknown	1056	X3-V19	216.58.206.163	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.35		C0:EA:E4:E4:FA:79
45	01.03.2018	15:00:19	10.10.22.15		unknown	1056	X3-V19	216.58.206.174	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.15		C0:EA:E4:E4:FA:87
46	01.03.2018	15:00:19	10.10.22.37		unknown	1056	X3-V19	92.45.114.212	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.37		C0:EA:E4:E4:FA:87
47	01.03.2018	15:00:19	10.10.22.35		unknown	1056	X3-V19	92.45.114.210	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.35		C0:EA:E4:E4:FA:65
48	01.03.2018	15:00:19	10.10.22.28		unknown	1056	X3-V19	104.24.21.50	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.28		C0:EA:E4:E4:FA:67
49	01.03.2018	15:00:20	10.10.22.9		unknown	1056	X3-V19	104.244.42.129	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.9:		C0:EA:E4:E4:FA:86
50	01.03.2018	15:00:20	10.10.22.32		unknown	1056	X3-V19	5.53.60.185	3260	X3-V19		0	0		tcp/3260		N/A			Possible TCP Flood on IF X3:V19 - src: 10.10.22.32		C0:EA:E4:E4:FA:A1

Şekil 4.2: Güvenlik Duvarı (Firewall) log kayıtları

4.2.2 Verinin düzenlenmesi ve temizlenmesi

Öncelikle 9 haftanın verisi tek dosyada birleştirilmiştir. Sonrasında ağ üzerinde iletişimi sağlayan hedef-kaynak port bilgileri, hedef-kaynak zone bilgileri, paket öncelik bilgisi, mesaj tipleri gibi alanlar, analizde kullanılmayacağı için temizlenmiştir.Ve verinin analiz edilmesi için daha uygun format olan XLS formatında alınıp, birleştirilen ve temizlenen loglar CSV formatına dönüştürülmüştür. Verilerin son hali Şekil 4.3 deki gibidir.

```
DATE;TIME;IP;MAC
01.03.2018;15:00:00;216.58.206.163;C0:EA:E4:E4:FA:86
01.03.2018;15:00:01;78.46.57.134;C0:EA:E4:E4:FA:72
01.03.2018;15:00:01;92.45.114.212;C0:EA:E4:E4:FA:A0
01.03.2018;15:00:02;172.217.17.174;C0:EA:E4:E4:FA:73
01.03.2018;15:00:04;96.4.64.79;C0:EA:E4:E4:FA:71
01.03.2018;15:00:04;40.85.190.15;C0:EA:E4:E4:FA:80
01.03.2018;15:00:05;92.45.114.212;C0:EA:E4:E4:FA:67
01.03.2018;15:00:05;196.196.140.0;C0:EA:E4:E4:FA:89
01.03.2018;15:00:05;66.220.158.11;C0:EA:E4:E4:FA:95
01.03.2018;15:00:06;92.45.114.212;C0:EA:E4:E4:FA:67
01.03.2018;15:00:06;20.113.35.170;C0:EA:E4:E4:FA:A1
01.03.2018;15:00:07;216.58.206.163;C0:EA:E4:E4:FA:82
01.03.2018;15:00:07;185.15.42.42;C0:EA:E4:E4:FA:99
01.03.2018;15:00:07;213.14.221.20;C0:EA:E4:E4:FA:83
01.03.2018;15:00:08;181.27.177.155;C0:EA:E4:E4:FA:65
01.03.2018;15:00:08;161.188.130.66;C0:EA:E4:E4:FA:83
01.03.2018;15:00:08;231.153.40.42;C0:EA:E4:E4:FA:79
01.03.2018;15:00:09;216.58.206.206;C0:EA:E4:E4:FA:73
01.03.2018;15:00:10;199.235.221.76;C0:EA:E4:E4:FA:84
01.03.2018;15:00:10;78.46.57.134;C0:EA:E4:E4:FA:72
01.03.2018;15:00:10;216.58.206.163;C0:EA:E4:E4:FA:79
01.03.2018;15:00:10;216.58.206.206;C0:EA:E4:E4:FA:66
01.03.2018;15:00:11;216.58.212.14;C0:EA:E4:E4:FA:86
01.03.2018;15:00:11;46.101.153.31;C0:EA:E4:E4:FA:74
01.03.2018;15:00:12;84.142.137.118;C0:EA:E4:E4:FA:76
01.03.2018;15:00:12;216.58.212.14;C0:EA:E4:E4:FA:78
01.03.2018;15:00:13;104.20.24.247;C0:EA:E4:E4:FA:A2
01.03.2018;15:00:13;216.58.206.174;C0:EA:E4:E4:FA:91
01.03.2018;15:00:13;78.46.57.134;C0:EA:E4:E4:FA:87
01.03.2018;15:00:13;216.58.206.163;C0:EA:E4:E4:FA:85
01.03.2018;15:00:14;216.58.206.163;C0:EA:E4:E4:FA:87
01.03.2018;15:00:14;172.217.17.174;C0:EA:E4:E4:FA:79
01.03.2018;15:00:14;179.119.207.112;C0:EA:E4:E4:FA:71
01.03.2018;15:00:14;85.111.19.140;C0:EA:E4:E4:FA:84
01.03.2018;15:00:15;66.220.158.11;C0:EA:E4:E4:FA:80
01.03.2018;15:00:16;109.169.55.249;C0:EA:E4:E4:FA:82
01.03.2018;15:00:16;104.20.59.198;C0:EA:E4:E4:FA:88
01.03.2018;15:00:16;92.45.114.210;C0:EA:E4:E4:FA:66
01.03.2018;15:00:17;92.45.114.210;C0:EA:E4:E4:FA:64
01.03.2018;15:00:18;94.199.203.214;C0:EA:E4:E4:FA:96
01.03.2018;15:00:18;34.196.124.157;C0:EA:E4:E4:FA:92
01.03.2018;15:00:18;216.58.206.206;C0:EA:E4:E4:FA:73
01.03.2018;15:00:18;216.58.206.163;C0:EA:E4:E4:FA:79
01.03.2018;15:00:19;216.58.206.174;C0:EA:E4:E4:FA:87
```

Şekil 4.3: Temizlenmiş ve birleştirilmiş log kayıtları(CSV formatı)

Şekil 4.3'deki işlenmiş veride 9 hafta içerisinde 71164 tane kayıt bulunmaktadır. Bu kayıtlar içerisinde tarih, saat, öğrenci MAC adresi, hedef IP bilgileri yer almaktadır.

4.2.2.1. R studio ortamında verilerin düzenlenmesi

Veri setinin R studio da birliktelik kuramı analizinde kullanılabilmesi için csv formatına dönüştürülen verinin matris yapısına dönüştürülmesi gerekmektedir. Bunun için öncelikle veriler okunur.

df_data ← read.csv("C:/Users/erdal/OneDrive/Masaüstü/tez.csv")

DATE: İşlem tarihi

TIME: İşlem saati

IP: Ziyaret edilen web sitesi

MAC: Öğrencilerin bilgisayarının fiziksel adresi

GEN: Cinsiyet

ORT: Mezuniyet Ortalaması

Diğer adımda veriler temizlenir ve düzenlenir.

R studio'da Apriori için gereken veriler aşağıdaki biçimde olmalıdır:

Tablo 4.1: Veri Düzenleme Örneği

IN	**Öğeler**
100	A C D
200	B C E
300	A B C E
400	B E

Tablo 4.1'de formatın, ilk sütunu her işlem için benzersiz bir tanımlayıcı içermelidir. İkinci sütun, bu işlemde yer alan, boşluk veya virgül veya başka bir ayırıcıyla ayrılmış öğelerden oluşur.

Elimizdeki veriler tablo 4.2'de gösterildiği gibidir.

Tablo 4.2: Veri setinin ilk görünümü

TARİH	MAC	IP
01.03.2018	C0:EA:E4:E4:FA:86	216.58.206.163
01.03.2018	C0:EA:E4:E4:FA:86	216.58.212.14
01.03.2018	C0:EA:E4:E4:FA:79	172.217.17.174
01.03.2018	C0:EA:E4:E4:FA:64	92.45.114.210

Verilerin yapısı ilişkilendirme kurallarını bulmak için gerekli formatta olmadığından, ilişkileri bulmadan önce bazı veri manipülasyonları yapılmalıdır.

Veri çerçevesini aynı günde aynı kişinin ziyaret ettiği web siteleri tek bir satırda olacak şekilde işlem biçimine dönüştürülmelidir. Bunun için, paket plyr tarafından sunulan, ddply adlı bir işlev kullanılır.

install.packages("plyr", dependencies= **TRUE**)

Oturuma ekli bir paketin olmadığından emin olunmalıdır. Hata almamak için önce dplyr paketini çıkartılmalı ardından paket tekrardan yüklenmelidir.

```
if(sessionInfo()['basePkgs']=="dplyr" | sessionInfo()['otherPkgs']=="dplyr"){
  detach(package:dplyr, unload=TRUE)
}
library(plyr)
```

Bir sonraki adım, öğrenci mac adresi ve işlem tarihini esas alarak veri çerçevesini ilgili formata dönüştürmektir.

Tablo 4.3: Veri setinin son görünümü

DATE	MAC	IP
01.03.2018	C0:EA:E4:E4:FA:86	216.58.206.163,216.58.212.14
01.03.2018	C0:EA:E4:E4:FA:79	172.217.17.174
01.03.2018	C0:EA:E4:E4:FA:64	92.45.114.210

01.03.2018 tarihinde C0:EA:E4:E4:FA:86 mac adresli öğrenci 216.58.206.163 ve 216.58.212.14 ip adreslerini ziyaret etmiştir. Aynı tarihte aynı öğrenci tarafından ziyaret edildikleri için bu IP adreslerini bir satırda virgülle ayrılarak birleştirilir.

Böylece Apriori bu veriler üzerinde uygulanabilir. Ddply işlevi daha büyük veri kümelerinde bile oldukça iyi çalışır.

İşlemleri yaptıktan sonra, analizde artık tarih ve diğer alanlara ihtiyaç yok. Bu sütunlar veri setinden temizlenmelidir.

df_itemList$Date ← **NULL**

df_itemList$Time ← **NULL**

df_itemList$Gen ← **NULL**

df_itemList$Ort ← **NULL**

df_itemList$Mac ← **NULL**

colnames(df_itemList) ← c("itemList")

Sonuçta ortaya çıkan tablo bir csv dosyasına yazılır. Bunu yapmamızın nedeni, bir .csv dosyasına bir veri çerçevesi yazarken, varsayılan olarak bir satır numarası eklemesidir.

write.csv(df_itemList,"C:/Users/erdal/OneDrive/ItemList.csv", quote = **FALSE**, row.names = **TRUE**)

Dosya içerisindeki IP adreslerine karşılık gelen hostnameler internetten araştırılarak bulunmuştur ve dosya buna göre revize edilmiştir. Dosya içeriği Şekil 4.4'de gösterildiği gibidir.

```
1,youtube.com,aliexpress.com
2,watsons.com.tr,sozcu.com.tr
3,gmail.com,ensonhaber.com,nesine.com
4,gmail.com,osym.gov.tr,reddit.com
5,youtube.com,Instagram.com,watsons.com.tr
6,aliexpress.com,sahibinden.com
7,gmail.com,facebook.com,sabah.com.tr,ensonhaber.com
8,facebook.com.tr,fotomac.com.tr,nesine.com
9,reddit.com,gratis.com,facebook.com
10,gmail.com,facebook.com.tr,Instagram.com,Twitter.com
11,sozcu.com.tr
12,beinsports.com,webrazzi.com
13,hepsiburada.com,sahibinden.com,kraloyun.com
14,facebook.com.tr,cozumpark.com,mshowto.org,gmail.com
15,gmail.com,ensonhaber.com,youtube.com,mshowto.org,forum.donanimhaber.com,ciscotr.com
16,youtube.com,chip.com.tr,kariyer.net
17,cozumpark.com,youtube.com,shiftdelete.net,gratis.com
```

Şekil 4.4: Veri setinin görünümü

4.2.3 Verilere bilgi eklenmesi

4.2.3.1. Anaconda ortamı için verilere bilgi eklenmesi

Çalışmada kullanılan laboratuvar içerisinde her öğrencinin kullandığı bilgisayarlar sabit ve kayıt altına alınmıştır. Her bilgisayarın MAC adresi öğrencilerle eşleştirilmiştir. Bu sayede Şekil 4.5'de görüldüğü üzere alınan log kayıtları içerisinde yer almayan cinsiyet sütunu oluşturulmuştur.

```
DATE;TIME;IP;MAC;GEN
01.03.2018;15:00:00;216.58.206.163;C0:EA:E4:E4:FA:86;M
01.03.2018;15:00:01;78.46.57.134;C0:EA:E4:E4:FA:72;M
01.03.2018;15:00:01;92.45.114.212;C0:EA:E4:E4:FA:A0;M
01.03.2018;15:00:02;172.217.17.174;C0:EA:E4:E4:FA:73;M
01.03.2018;15:00:04;96.4.64.79;C0:EA:E4:E4:FA:71;M
01.03.2018;15:00:04;40.85.190.15;C0:EA:E4:E4:FA:80;F
01.03.2018;15:00:05;92.45.114.212;C0:EA:E4:E4:FA:67;F
01.03.2018;15:00:05;196.196.140.0;C0:EA:E4:E4:FA:89;M
01.03.2018;15:00:05;66.220.158.11;C0:EA:E4:E4:FA:95;F
01.03.2018;15:00:06;92.45.114.212;C0:EA:E4:E4:FA:67;F
01.03.2018;15:00:06;20.113.35.170;C0:EA:E4:E4:FA:A1;M
01.03.2018;15:00:07;216.58.206.163;C0:EA:E4:E4:FA:82;M
01.03.2018;15:00:07;185.15.42.42;C0:EA:E4:E4:FA:99;F
01.03.2018;15:00:07;213.14.221.20;C0:EA:E4:E4:FA:83;M
01.03.2018;15:00:08;181.27.177.155;C0:EA:E4:E4:FA:65;M
01.03.2018;15:00:08;161.188.130.66;C0:EA:E4:E4:FA:83;M
01.03.2018;15:00:08;231.153.40.42;C0:EA:E4:E4:FA:79;M
01.03.2018;15:00:09;216.58.206.206;C0:EA:E4:E4:FA:73;M
01.03.2018;15:00:10;199.235.221.76;C0:EA:E4:E4:FA:84;M
01.03.2018;15:00:10;78.46.57.134;C0:EA:E4:E4:FA:72;M
01.03.2018;15:00:10;216.58.206.163;C0:EA:E4:E4:FA:79;M
01.03.2018;15:00:10;216.58.206.206;C0:EA:E4:E4:FA:66;M
01.03.2018;15:00:11;216.58.212.14;C0:EA:E4:E4:FA:86;M
01.03.2018;15:00:11;46.101.153.31;C0:EA:E4:E4:FA:74;M
01.03.2018;15:00:12;84.142.137.118;C0:EA:E4:E4:FA:76;M
01.03.2018;15:00:12;216.58.212.14;C0:EA:E4:E4:FA:78;M
01.03.2018;15:00:13;104.20.24.247;C0:EA:E4:E4:FA:A2;M
01.03.2018;15:00:13;216.58.206.174;C0:EA:E4:E4:FA:91;M
01.03.2018;15:00:13;78.46.57.134;C0:EA:E4:E4:FA:87;M
01.03.2018;15:00:13;216.58.206.163;C0:EA:E4:E4:FA:85;F
01.03.2018;15:00:14;216.58.206.163;C0:EA:E4:E4:FA:87;M
01.03.2018;15:00:14;172.217.17.174;C0:EA:E4:E4:FA:79;M
01.03.2018;15:00:14;179.119.207.112;C0:EA:E4:E4:FA:71;M
01.03.2018;15:00:14;85.111.19.140;C0:EA:E4:E4:FA:84;M
01.03.2018;15:00:15;66.220.158.11;C0:EA:E4:E4:FA:80;F
01.03.2018;15:00:16;109.169.55.249;C0:EA:E4:E4:FA:82;M
01.03.2018;15:00:16;104.20.59.198;C0:EA:E4:E4:FA:88;M
01.03.2018;15:00:16;92.45.114.210;C0:EA:E4:E4:FA:66;M
01.03.2018;15:00:17;92.45.114.210;C0:EA:E4:E4:FA:64;F
01.03.2018;15:00:18;94.199.203.214;C0:EA:E4:E4:FA:96;M
01.03.2018;15:00:18;34.196.124.157;C0:EA:E4:E4:FA:92;M
01.03.2018;15:00:18;216.58.206.206;C0:EA:E4:E4:FA:73;M
01.03.2018;15:00:18;216.58.206.163;C0:EA:E4:E4:FA:79;M
01.03.2018;15:00:19;216.58.206.174;C0:EA:E4:E4:FA:87;M
```

Şekil 4.5: Cinsiyet sütunu eklenmiş log kayıtları

Şekil 4.6'da ise MAC adresleriyle eşleştirilen öğrencilerin ilgili dersteki ders notu log kayıtlarına eklenmiştir. Eklenen ders notu öğrencinin yıl içerisinde aldığı vize(%40) ve final(%60) notunun ortalaması ile hesaplanmış notlardır. Bu notlar okulun ois veritabanından alınarak log kaydına ilave edilmiştir.

```
DATE,TIME,IP,MAC,GEN,ORT
01.03.2018,15:00:01,78.46.57.134,C0:EA:E4:E4:FA:72,M,66.4
01.03.2018,15:00:04,40.85.190.15,C0:EA:E4:E4:FA:80,F,15.2
01.03.2018,15:00:05,66.220.158.11,C0:EA:E4:E4:FA:95,F,64.53
01.03.2018,15:00:07,185.15.42.42,C0:EA:E4:E4:FA:99,F,82.26
01.03.2018,15:00:07,213.14.221.20,C0:EA:E4:E4:FA:83,M,56.13
01.03.2018,15:00:09,216.58.206.206,C0:EA:E4:E4:FA:73,M,53.8
01.03.2018,15:00:10,78.46.57.134,C0:EA:E4:E4:FA:72,M,66.4
01.03.2018,15:00:10,216.58.206.206,C0:EA:E4:E4:FA:66,M,59.4
01.03.2018,15:00:11,46.101.153.31,C0:EA:E4:E4:FA:74,M,63.13
01.03.2018,15:00:13,104.20.24.247,C0:EA:E4:E4:FA:A2,M,71.06
01.03.2018,15:00:13,216.58.206.174,C0:EA:E4:E4:FA:91,M,58.23
01.03.2018,15:00:13,78.46.57.134,C0:EA:E4:E4:FA:87,M,55.2
01.03.2018,15:00:14,85.111.19.140,C0:EA:E4:E4:FA:84,M,61.73
01.03.2018,15:00:15,66.220.158.11,C0:EA:E4:E4:FA:80,F,15.2
01.03.2018,15:00:16,109.169.55.249,C0:EA:E4:E4:FA:82,M,73.86
01.03.2018,15:00:16,104.20.59.198,C0:EA:E4:E4:FA:88,M,96.2
01.03.2018,15:00:18,34.196.124.157,C0:EA:E4:E4:FA:92,M,64.3
01.03.2018,15:00:18,216.58.206.206,C0:EA:E4:E4:FA:73,M,53.8
01.03.2018,15:00:19,216.58.206.174,C0:EA:E4:E4:FA:87,M,55.2
01.03.2018,15:00:19,104.24.21.50,C0:EA:E4:E4:FA:67,F,61.26
01.03.2018,15:00:20,104.244.42.129,C0:EA:E4:E4:FA:86,M,70.83
01.03.2018,15:00:21,85.111.48.117,C0:EA:E4:E4:FA:81,M,22.26
01.03.2018,15:00:23,198.11.132.250,C0:EA:E4:E4:FA:69,M,80.16
01.03.2018,15:00:24,34.196.124.157,C0:EA:E4:E4:FA:85,F,94.06
01.03.2018,15:00:24,216.58.206.206,C0:EA:E4:E4:FA:66,M,59.4
01.03.2018,15:00:25,104.20.24.247,C0:EA:E4:E4:FA:94,M,68.5
01.03.2018,15:00:27,217.74.24.159,C0:EA:E4:E4:FA:81,M,22.26
01.03.2018,15:00:28,46.45.154.80,C0:EA:E4:E4:FA:82,M,73.86
01.03.2018,15:00:29,213.14.221.20,C0:EA:E4:E4:FA:67,F,61.26
01.03.2018,15:00:33,46.101.153.31,C0:EA:E4:E4:FA:93,M,81.46
01.03.2018,15:00:33,216.58.206.174,C0:EA:E4:E4:FA:80,F,15.2
01.03.2018,15:00:35,34.196.124.157,C0:EA:E4:E4:FA:92,M,64.3
01.03.2018,15:00:37,216.58.206.206,C0:EA:E4:E4:FA:87,M,55.2
01.03.2018,15:00:38,46.101.153.31,C0:EA:E4:E4:FA:97,M,69.66
01.03.2018,15:00:40,104.20.24.247,C0:EA:E4:E4:FA:64,F,56.83
01.03.2018,15:00:40,185.25.101.167,C0:EA:E4:E4:FA:87,M,55.2
01.03.2018,15:00:41,104.20.59.198,C0:EA:E4:E4:FA:65,M,63.13
01.03.2018,15:00:44,176.53.43.5,C0:EA:E4:E4:FA:77,M,84.83
01.03.2018,15:00:46,185.25.101.167,C0:EA:E4:E4:FA:75,M,54.06
01.03.2018,15:00:47,85.111.48.117,C0:EA:E4:E4:FA:80,F,15.2
01.03.2018,15:00:48,85.111.48.117,C0:EA:E4:E4:FA:80,F,15.2
01.03.2018,15:00:48,157.240.20.35,C0:EA:E4:E4:FA:95,F,64.53
01.03.2018,15:00:49,216.58.212.5,C0:EA:E4:E4:FA:73,M,53.8
01.03.2018,15:00:52,85.111.19.140,C0:EA:E4:E4:FA:64,F,56.83
```

Şekil 4.6: Mezuniyet ortalaması sütunu eklenmiş log kayıtları

4.3 Verinin Modellenmesi

4.3.1. Anaconda ortamında verinin modellenmesi

İşlenmiş veriler Python Anaconda çalışma ortamı kullanılarak algoritmanın kullanımına uygun hale getirilmiştir. Böylece verinin modellenmesi sağlanmıştır. Çalışma ortamında kullanılan kütüphane bilgileri Şekil 4.7'de gösterilmiştir.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
from pyspark.ml.fpm import FPGrowth
from pyspark.sql.functions import collect_list, col
import random
```

Şekil 4.7: Anaconda çalışma ortamında kullanılan kütüphaneler

Şekil 0.2-4.7: Anaconda çalışma ortamında kullanılan kütüphaneler

Şekil 4.8 deki kod bloğu anaconda çalışma ortamında işlenmiş log kayıtların program içerisinde okunmasını sağlamaktadır.

```
data = pd.read_csv('Tez.csv')
```

Şekil 4.8: Verilerin okunması

Şekil 0.3-4.8: Verilerin okunması

Veriler içerisindeki hedef IP bilgilerine karşılık gelen hostname ler internetten araştırılarak (ipsorgu.com sitesinden IP adreslerinin kayıtlı host adresleri tespit edilmiştir.) bulunmuştur. Bulunan host adresleri anaconda çalışma ortamında yazılan fonksiyon aracılığı ile veri setine hostname kolonu olarak çalışma zamanında eklenmiştir. Şekil 4.9'da yazılan fonksiyon hangi IP adresinin hangi host adresine karşılık geldiğini döndürür. Şekil 4.10 ise hostname kolonuna ilgili host adresinin atanması için yazılan kod bloğunu göstermektedir.

```
def set_hostname(IP):
    if IP == '66.220.158.11' or IP == '157.240.20.15' or IP == '157.240.20.35' :
        return 'Facebook'
    if IP == '104.244.42.129' :
        return 'Twitter'
    if IP == '34.196.124.157' :
        return 'Instagram'
    if IP == '216.58.206.206' or IP == '216.58.206.174' :
        return 'Youtube'
    if IP == '104.20.24.247' :
        return 'Onedio'
    if IP == '176.53.43.5' :
        return 'Ekşi Sözlük'
    if IP == '104.20.59.198' :
        return 'Ensonhaber'
    if IP == '78.46.57.134' :
        return 'Sözcü'
    if IP == '109.169.55.249' :
        return 'Sabah'
    if IP == '46.101.153.31' :
        return 'Fotomaç'
    if IP == '85.111.19.140' :
        return 'Trendyol'
    if IP == '193.28.225.200' :
        return 'Hepsiburada'
    if IP == '40.85.190.15' :
        return 'Gratis'
    if IP == '85.111.48.117' :
        return 'Watsons'
    if IP == '104.24.21.50' :
        return 'Webrazzi'
    if IP == '85.111.30.111' :
        return 'Sahibinden'
    if IP == '172.217.17.174' or IP == '216.58.206.163' or IP == '216.58.212.14':
        return 'Google'
    if IP == '46.253.112.23' :
        return 'Kral Oyun'
    if IP == '94.199.203.214' or IP == '92.45.114.212' or IP == '92.45.114.210':
        return 'Ayvansaray OIS'
    if IP == '216.58.212.5' :
        return 'Gmail'
    if IP == '151.101.65.140' :
        return 'Reddit'
    if IP == '104.25.193.37' :
```

Şekil 4.9: Host adreslerinin tanımlanması

```
data['HOSTNAME'] = data['IP'].apply(set_hostname)
```

Şekil 4.10: Host adreslerinin atamasının yapılması

Düzenlenmiş verinin son hali Şekil 4.11'de gösterildiği gibidir.

	DATE	TIME	IP	MAC	GEN	ORT	HOSTNAME	GROUP
0	01.03.2018	15:00:01	78.46.57.134	C0:EA:E4:E4:FA:72	M	66.40	Sözcü	Haber Siteleri
1	01.03.2018	15:00:04	40.85.190.15	C0:EA:E4:E4:FA:80	F	15.20	Gratis	Alış Veriş Siteleri
2	01.03.2018	15:00:05	66.220.158.11	C0:EA:E4:E4:FA:95	F	64.53	Facebook	Sosyal Medya
3	01.03.2018	15:00:07	185.15.42.42	C0:EA:E4:E4:FA:99	F	82.26	Chip	Bilişim
4	01.03.2018	15:00:07	213.14.221.20	C0:EA:E4:E4:FA:83	M	56.13	osym.gov.tr	Kariyer
5	01.03.2018	15:00:09	216.58.206.206	C0:EA:E4:E4:FA:73	M	53.80	Youtube	Sosyal Medya
6	01.03.2018	15:00:10	78.46.57.134	C0:EA:E4:E4:FA:72	M	66.40	Sözcü	Haber Siteleri
7	01.03.2018	15:00:10	216.58.206.206	C0:EA:E4:E4:FA:66	M	59.40	Youtube	Sosyal Medya
8	01.03.2018	15:00:11	46.101.153.31	C0:EA:E4:E4:FA:74	M	63.13	Fotomaç	Spor
9	01.03.2018	15:00:13	104.20.24.247	C0:EA:E4:E4:FA:A2	M	71.06	Onedio	Sosyal Medya
10	01.03.2018	15:00:13	216.58.206.174	C0:EA:E4:E4:FA:91	M	58.23	Youtube	Sosyal Medya
11	01.03.2018	15:00:13	78.46.57.134	C0:EA:E4:E4:FA:87	M	55.20	Sözcü	Haber Siteleri
12	01.03.2018	15:00:14	85.111.19.140	C0:EA:E4:E4:FA:84	M	61.73	Trendyol	Alış Veriş Siteleri
13	01.03.2018	15:00:15	66.220.158.11	C0:EA:E4:E4:FA:80	F	15.20	Facebook	Sosyal Medya
14	01.03.2018	15:00:16	109.169.55.249	C0:EA:E4:E4:FA:82	M	73.86	Sabah	Haber Siteleri
15	01.03.2018	15:00:16	104.20.59.198	C0:EA:E4:E4:FA:88	M	96.20	Ensonhaber	Haber Siteleri
16	01.03.2018	15:00:18	34.196.124.157	C0:EA:E4:E4:FA:92	M	64.30	Instagram	Sosyal Medya
17	01.03.2018	15:00:18	216.58.206.206	C0:EA:E4:E4:FA:73	M	53.80	Youtube	Sosyal Medya
18	01.03.2018	15:00:19	216.58.206.174	C0:EA:E4:E4:FA:87	M	55.20	Youtube	Sosyal Medya
19	01.03.2018	15:00:19	104.24.21.50	C0:EA:E4:E4:FA:67	F	61.26	Webrazzi	Haber Siteleri
20	01.03.2018	15:00:20	104.244.42.129	C0:EA:E4:E4:FA:86	M	70.83	Twitter	Sosyal Medya
21	01.03.2018	15:00:21	85.111.48.117	C0:EA:E4:E4:FA:81	M	22.26	Watsons	Alış Veriş Siteleri
22	01.03.2018	15:00:23	198.11.132.250	C0:EA:E4:E4:FA:69	M	80.16	aliexpress.com	Alış Veriş Siteleri
23	01.03.2018	15:00:24	34.196.124.157	C0:EA:E4:E4:FA:85	F	94.06	Instagram	Sosyal Medya
24	01.03.2018	15:00:24	216.58.206.206	C0:EA:E4:E4:FA:66	M	59.40	Youtube	Sosyal Medya
25	01.03.2018	15:00:25	104.20.24.247	C0:EA:E4:E4:FA:94	M	68.50	Onedio	Sosyal Medya
26	01.03.2018	15:00:27	217.74.24.159	C0:EA:E4:E4:FA:81	M	22.26	Nesine.com	Spor
27	01.03.2018	15:00:28	46.45.154.80	C0:EA:E4:E4:FA:82	M	73.86	Forum Donanım Haber	Bilişim
28	01.03.2018	15:00:29	213.14.221.20	C0:EA:E4:E4:FA:67	F	61.26	osym.gov.tr	Kariyer
29	01.03.2018	15:00:33	46.101.153.31	C0:EA:E4:E4:FA:93	M	81.46	Fotomaç	Spor

Şekil 4.11: Host adresleri eklenmiş veri seti

4.3.2. R Studio ortamında verinin modellenmesi

R studio için hazırlanan dosya okunur. Dosya içeriği Şekil 4.4'de gösterildiği gibidir.

dataset = read.csv('C:/Users/erdal/OneDrive/ItemList.csv', header = ***FALSE****)*

Öğeler arasındaki ilişkileri bulmak için ItemList.csv'de algoritma çalıştırılır. Apriori, bu ilişkileri birlikte aynı gün aynı mac adresine sahip öğrencilerin gittiği sitelerin sıklığına dayanarak bulur.

R studio daki uygulama için, işlemleri okumak ve ilişkilendirme kurallarını bulmak için işlevler sağlayan 'arules' adlı bir paket bulunmaktadır.

*install.packages("arules", dependencies=****TRUE****)*

library*(arules)*

Read.transactions() işlevlerini kullanarak, ItemList.csv dosyasını okunur ve bir işlem biçimine dönüştürülür.

dataset = read.transactions('C:/Users/erdal/OneDrive/ItemList.csv', sep=',', rm.duplicates = ***TRUE)***

Parametreler: İşlem dosyası: ItemList.csv

rm.duplicates: Yinelenen hiçbir işlem girmediğimizden emin olmak için kullanılır.

Format: satır 1: işlem kimlikleri, satır 2: öğe listesi

sep: Öğeler arasında ayırıcı, bu durumda virgül

cols: İşlem kimlikleri için sütun sayısı

Son olarak, destek ve güven için asgari değerler belirlenerek işlemlerde apriori algoritmasını çalıştırılır.

rules ← apriori(dataset,parameter = list(sup = ***0.01****, conf =* ***0.5****,target="rules"));*

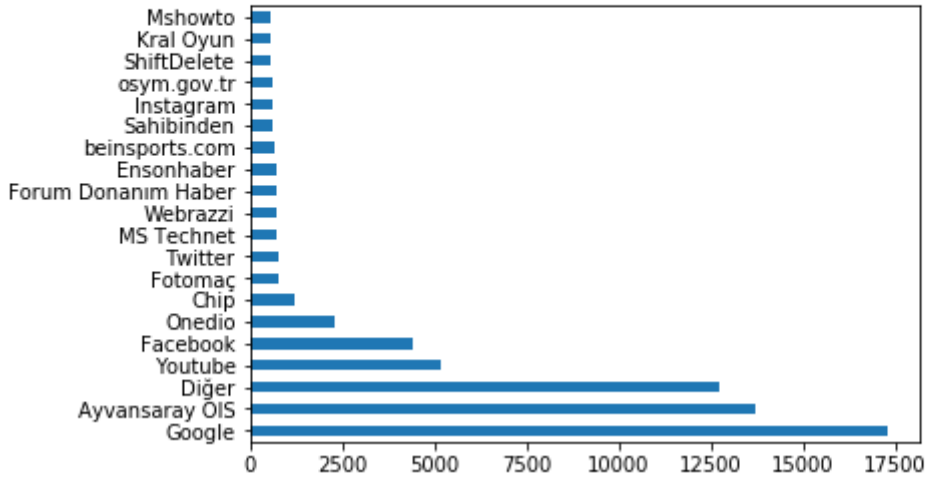
4.4 Modelin Değerlendirilmesi

4.4.1 Anaconda geliştirme ortamında verilerin değerlendirilmesi

En sık ziyaret edilen 20 site Şekil 4.12'de görüldüğü gibidir. Grafikte de görüldüğü üzere Ayvansaray OIS internet sitesi, browserların açılış sayfası olduğundan dolayı ziyaret sayısı miktarca fazla olarak gözlemlenmiştir. Bu nedenle elimine edilecektir. Google için de aynı durum söz konusudur.

```
data['HOSTNAME'].value_counts()[:20].plot(kind='barh')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0xd67fcf8>
```

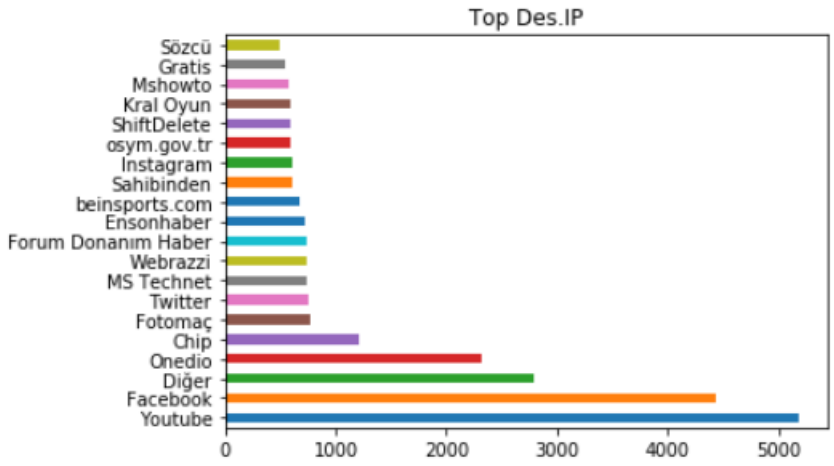


Şekil 4.12: Sık ziyaret edilen 20 website

Arama siteleri ve açılış sayfaları olan Ayvansaray internet siteleri elimine edildikten sonra en çok ziyaret edilen 20 site Şekil 4.13'de görüldüğü gibi sıralanmıştır.

```
data['HOSTNAME'].value_counts()[:20].plot(kind='barh', title='Top Des.IP')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x874151fa90>
```

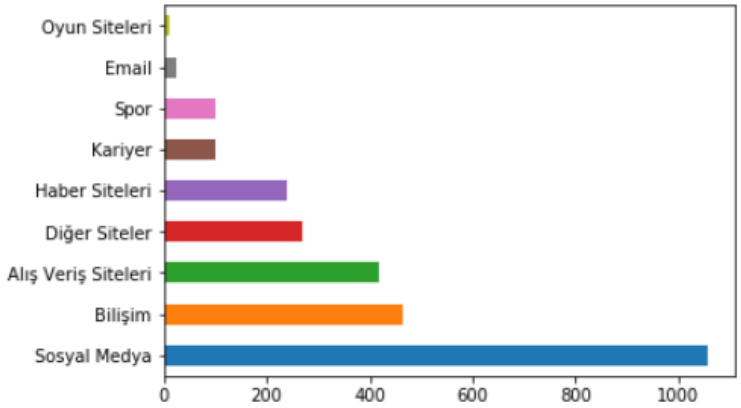


Şekil 4.13: Eleme işleminden sonra sık ziyaret edilen 20 website

Yapılan işlem ile gidilen sitelerin içerik bilgilerine göre gruplama yapılmıştır. Gruplama adımlarında Şekil 4.14'de görüldüğü üzere gidilen sitelerin Oyun, E-mail, Spor, Kariyer, Haber, Alış Veriş, Bilişim, Sosyal Medya ve diğer isimli gruplar ile içerikler gruplandırılmıştır.

```
datax['GROUP'].value_counts().plot(kind='barh')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x8741573240>
```

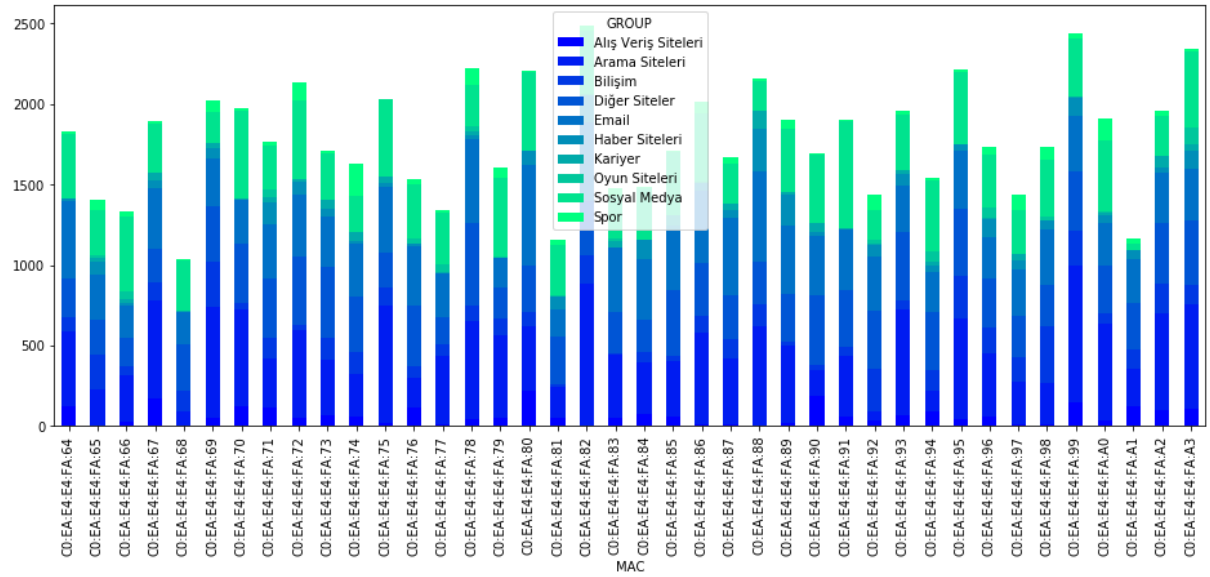


Şekil 4.14: Websitelerin gruplandırılması

Şekil 4.15'de ziyaret edilen web siteleri gruplandırılarak her öğrencinin hangi grupta yer alan sitelere daha fazla ziyarette bulunduğu saptanmıştır.

```
count_websites_by_mac.plot(kind='bar', stacked=True, figsize=[16,6], colormap='winter')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1071eb70>
```

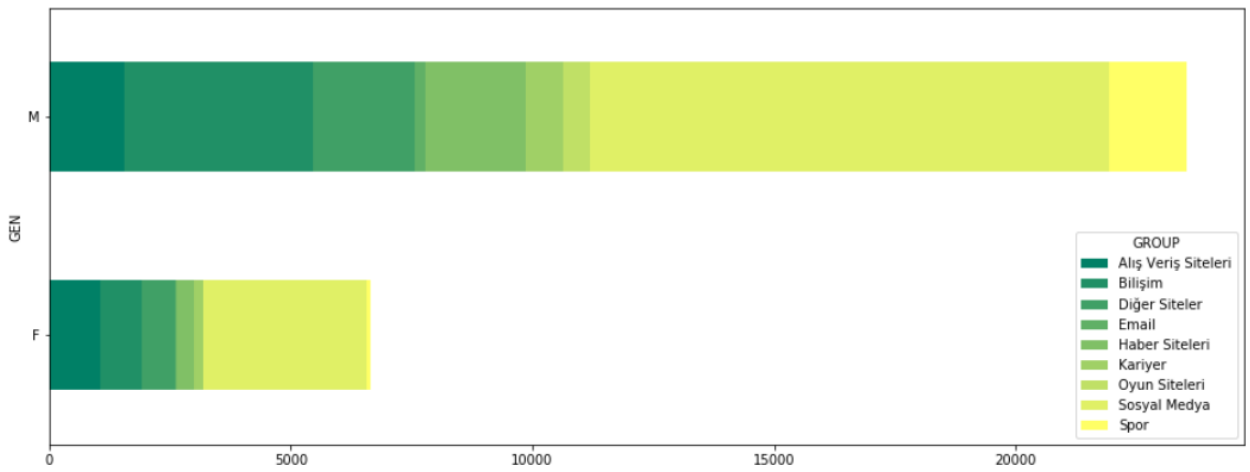


Şekil 4.15: Öğrencilerin ziyaret ettiği website grupları

Şekil 4.16'da gidilen sitelerin cinsiyetlere göre dağılımı ise aşağıdaki gibidir. Grafikte görüldüğü üzere her iki cinsiyetteki öğrencilerin ilk tercihlerinin sosyal medya içerikli siteler olduğu görülmektedir. Sosyal medya içerikli websitelerinden sonra en çok tercih edilen sitelerin, erkek öğrencilerde bilişim siteleri oluştururken, kadın öğrencilerde alış veriş siteleri olduğu görülmektedir.

```
count_websites_by_Gen.plot(kind='barh', stacked=True, figsize=[16,6], colormap='summer')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x89b3773e48>
```



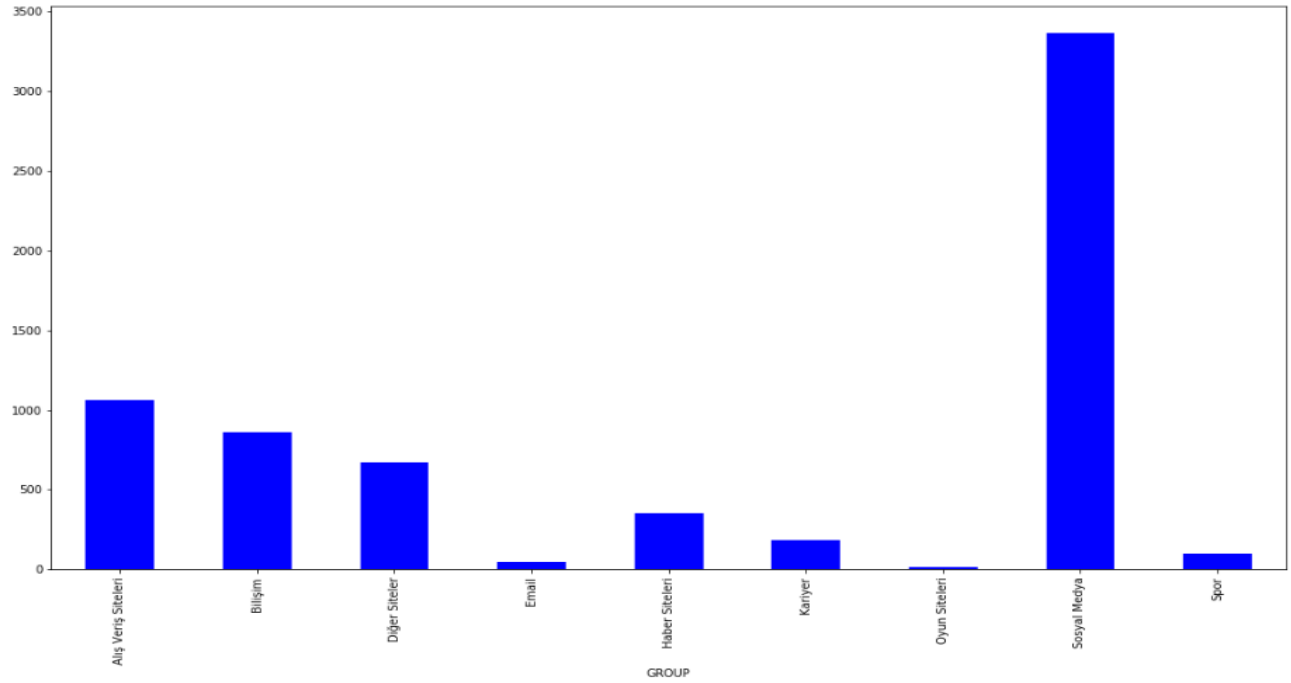
Şekil 4.16: Cinsiyete göre website gruplarının dağılımı

Öğrencilerin gittiği siteler belli gruplara ayrılmıştır. Bu gruplar Şekil 4.17'de ve Şekil 4.18'de görülmektedir. Bu gruplar Alış Veriş Siteleri, Bilişim, Diğer Siteler, Email, Haber Siteleri, Kariyer, Oyun Siteleri, Sosyal Medya, Spor olarak görülmektedir. Şekil 4.17'de kadınların gittiği website grupların genel görünümü görülmektedir.

Kadın öğrencilerin en çok sosyal medya sitelerine gitttikleri daha sonrasında alışveriş sitelerine ilgi gösterdikleri görünmektedir. Erkeklerin gittikleri website grupları Şekil 4.18'de görünmektedir. Erkek öğrencilerinde kadın öğrenciler gibi ilk tercihleri sosyal medya içerikli websiteleri olduğu tespit edilmiştir. Sosyal medya içeriğinden sonra en çok bilişim sitelerine gittikleri görünmüştür. Ayrıca erkek öğrencilerin spor içerikli websitelere kadın öğrencilere göre daha yoğun eriştikleri görünmektedir. Grafiklerde görülmek üzere diğer sitelere erişim oranları erkeklerde de kadınlarda aynı olduğu tespit edilmiştir.

```
datav2 = (data[(data['GEN'] == 'F')]
          .groupby(['GROUP'])
         .size())
datav2
datav2.plot(kind='bar', stacked=True, figsize=[18,10], colormap='winter')
```

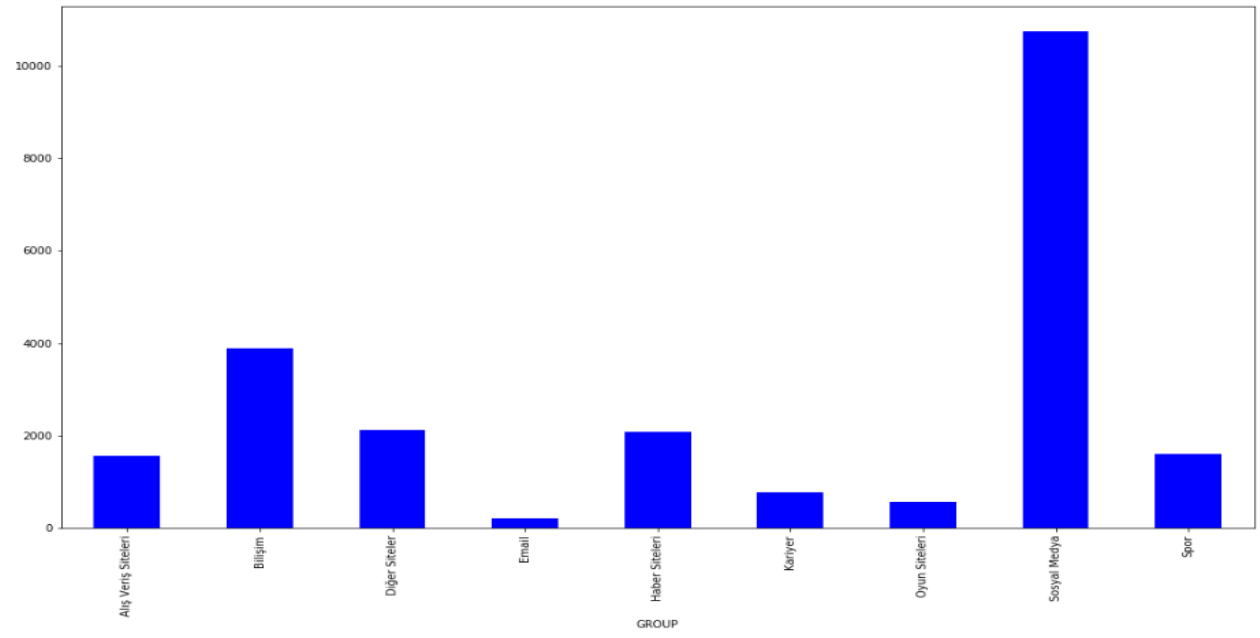
```
<matplotlib.axes._subplots.AxesSubplot at 0x7d7288a20>
```



Şekil 4.17: Kadınların gittiği website grupları genel görünümü

```
datav2 = (data[(data['GEN'] == 'M')]
          .groupby(['GROUP'])
         .size())
datav2
datav2.plot(kind='bar', stacked=True, figsize=[18,10], colormap='winter')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7d75c8940>
```

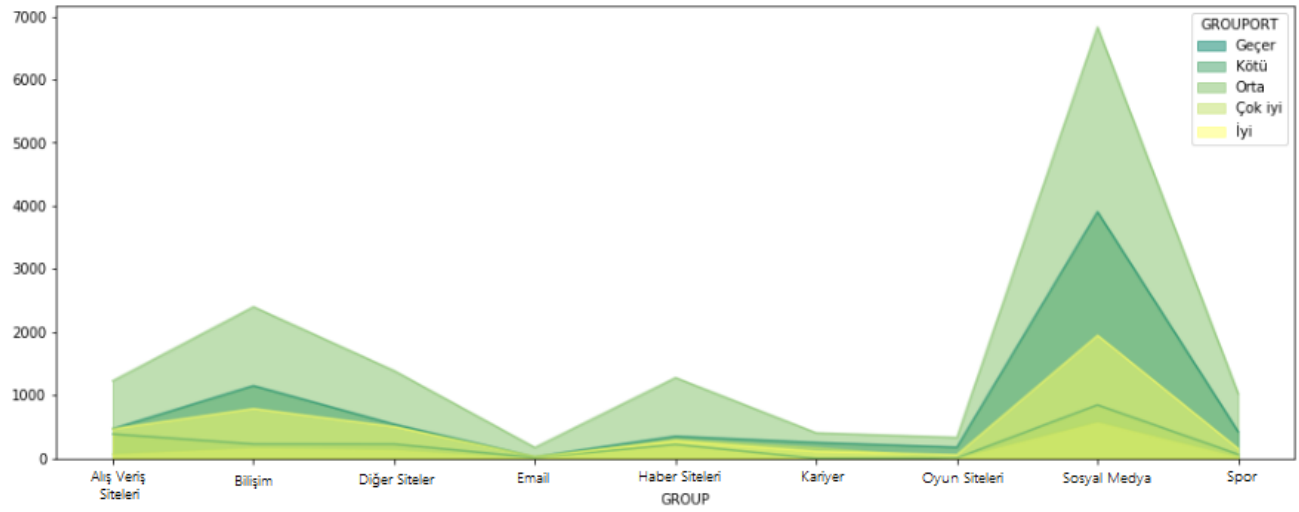


Şekil 4.18: Erkeklerin gittiği website grupları genel görünümü

Öğrencilerin ders esnasında eriştikleri web sitelerin, ilgili dersteki aldıkları ders ortalama notuna göre çıkarılmış grafik Şekil 4.19'da görünmektedir. Şekil 4.19'da beş adet renk skalasında mevcuttur. Bu skala Tablo 4.4'de görüldüğü üzere okulun ders geçme ve harfle gösterim yapısına göre kategorize edilmiştir. Şekil 4.19'da görüldüğü üzere iyi ve çok iyi notla geçen öğrencilerin oyun sitelerine az gittiği görünmektedir. Kötü olarak nitelendirilen ve ilgili senede dersten kalan öğrencilerin daha çok haber sitelerine eriştikleri tespit edilmiştir.

```
count_websites_by_Ort.plot(kind='area', stacked=False, figsize=[16,6], colormap='summer')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7d9012ba8>
```



Şekil 4.19: Ders notu ortalamalarına göre website gruplarının dağılımı

Tablo 4.4: Ders Notu Çizelgesi

Notla Gösterim	Harfle Gösterim	Başarı Kriteri
90-100	AA	Çok iyi
85-90	BA	İyi
80-84	BB	İyi
70-79	CB	Orta
60-69	CC	Orta
55-59	DC	Geçer
50-54	DD	Geçer
35-49	FD	Kötü
0-34	FF	Kötü

Şekil 4.20'de ders notu 50'den az not alan kadın öğrencilerin kod bloğu ve matris yapısı görünmektedir. Şekil 4.21'de ders notu 50'den az not alan erkek öğrencilerin kod bloğu ve matris yapısı görünmektedir. Bu matris yapısı ile 50'den düşük not alan öğrencilerin gittikleri web sitelerin ve bu değerlere bağlı olarak sosyal medya içerikli grafiklerin çıkarılmasına referans olmaktadır.

```
datav2 = (data[(data['GEN'] == 'F') & (data['ORT'] < 50.00)].groupby(['GROUP','HOSTNAME'])
          .size().unstack()
          .reset_index().fillna(0)
          .set_index('GROUP'))
datav2
```

HOSTNAME	Chip	Ciscotr	Diğer	Ekşi Sözlük	Facebook	Forum Donanım Haber	Fotomaç	Gmail	Gratis	Instagram	...	ShiftDelete	Tamindir.com	Trendyol	Twitter	Watsons
GROUP																
Alış Veriş Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	150.0	0.0	...	0.0	0.0	3.0	0.0	61.0
Bilişim	17.0	4.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	...	6.0	9.0	0.0	0.0	0.0
Diğer Siteler	0.0	0.0	85.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
Email	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
Haber Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
Kariyer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
Oyun Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
Sosyal Medya	0.0	0.0	0.0	2.0	200.0	0.0	0.0	0.0	0.0	3.0	...	0.0	0.0	0.0	1.0	0.0
Spor	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0

9 rows × 26 columns

Şekil 4.20: Ders notu 50'den küçük kadın öğrencilerin matris görünümü

```
datav2 = (data[(data['GEN'] == 'M') & (data['ORT'] < 50.00)].groupby(['GROUP','HOSTNAME'])
          .size().unstack()
          .reset_index().fillna(0)
          .set_index('GROUP'))
datav2
```

HOSTNAME	Chip	Ciscotr	Diğer	Ekşi Sözlük	Facebook	Forum Donanım Haber	Fotomaç	Gmail	Gratis	Instagram	...	ShiftDelete	Sözcü	Tamindir.com	Twitter	Watsons	\
GROUP																	
Alış Veriş Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	19.0	0.0	...	0.0	0.0	0.0	0.0	32.0	
Bilişim	116.0	1.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	...	11.0	0.0	3.0	0.0	0.0	
Diğer Siteler	0.0	0.0	142.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
Email	0.0	0.0	0.0	0.0	0.0	0.0	0.0	12.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
Haber Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	56.0	0.0	0.0	0.0	
Kariyer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
Oyun Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	
Sosyal Medya	0.0	0.0	0.0	37.0	21.0	0.0	0.0	0.0	0.0	3.0	...	0.0	0.0	0.0	2.0	0.0	
Spor	0.0	0.0	0.0	0.0	0.0	0.0	26.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	

9 rows × 27 columns

Şekil 4.21: Ders notu 50'den küçük erkek öğrencilerin matris görünümü

Şekil 4.22’de 50’den az not alana ve kötü olarak kategorize edilen kadın öğrencilerin erişim sağladıkları web siteler yeralmaktadır.Grafiğin sağ alt kısmında renk skalasıyla grupların içinde, ilgili gruba ait web siteler görünmektedir. Kadınların alış veriş sitelerinde turuncu olarak görülen oranda gratis sitesini görmekteyiz. Gratisten sonra en yoğun erişim sağlanan alışveriş sitesinin Trendyol olduğu tespit edilmiştir.

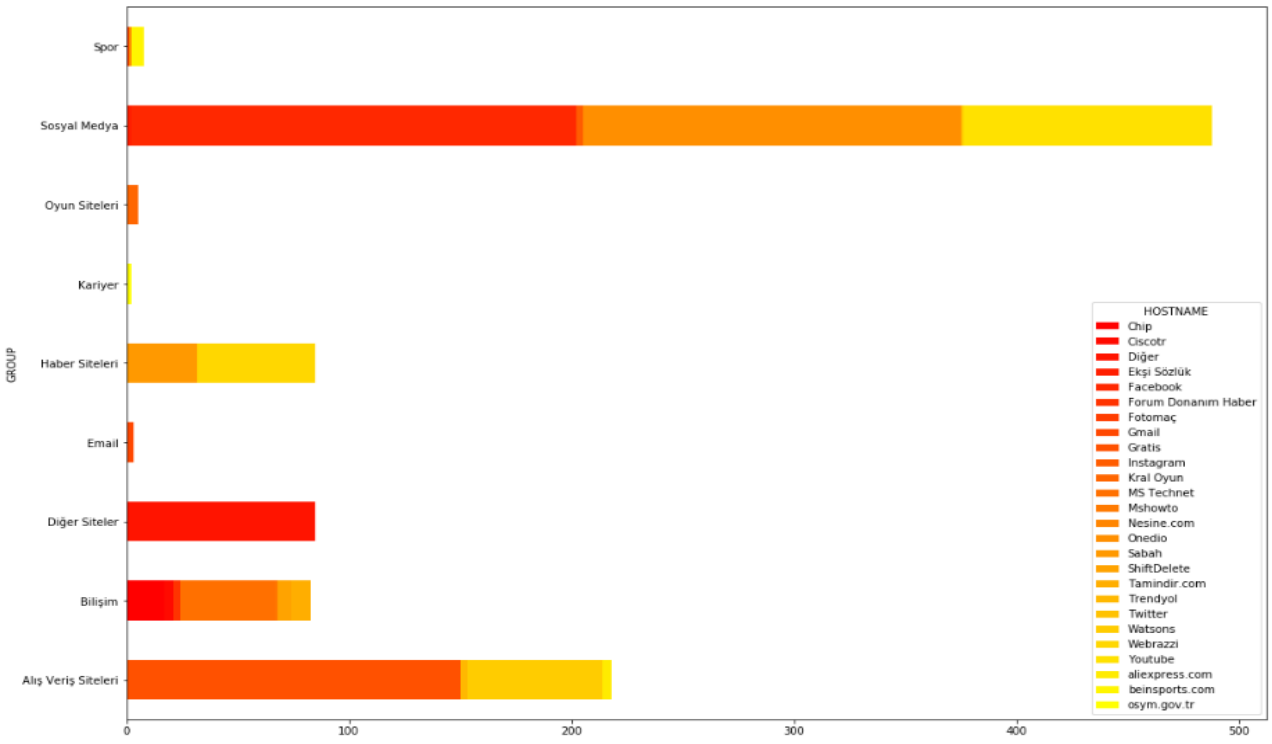
Şekil 4.23’de 50’den az not alana ve kötü olarak kategorize edilen erkek öğrencilerin erişim sağladıkları web siteler yeralmaktadır. Erkek öğrencilerin kadınlara kıyasla haber siteleri ve bilişim sitelerine daha çok eriştikleri görünmektedir.

Şekil 4.24’de 50’den düşük not almış kadın öğrencilerin sosyal medya içerikli sitelere erişimleri yeralmaktadır. Şeklin üst kısmında sosyal medya içeriklerinin filtrelenmesini sağlayan kod bloğu yeralmaktadır. Kadın öğrencilerin en çok erişim sağladıkları sosyal medya sitesinin facebook ve onedio olarak görünmektedir.

Şekil 4.25’de 50’den düşük not almış erkek öğrencilerin sosyal medya içerikli sitelere erişimleri yeralmaktadır. Şeklin üst kısmında sosyal medya içeriklerinin filtrelenmesini sağlayan kod bloğu yeralmaktadır. Erkek öğrencilerin en çok erişim sağladıkları sosyal medya sitesinin reddit ve youtube olduğu görünmektedir. Ayrıca erkeklerin kadınlara göre ekşi sözlük sitesine daha çok eriştikleri tespit edilmiştir.

```
datav2.plot(kind='barh', stacked=True, figsize=[18,13], colormap='autumn')
```

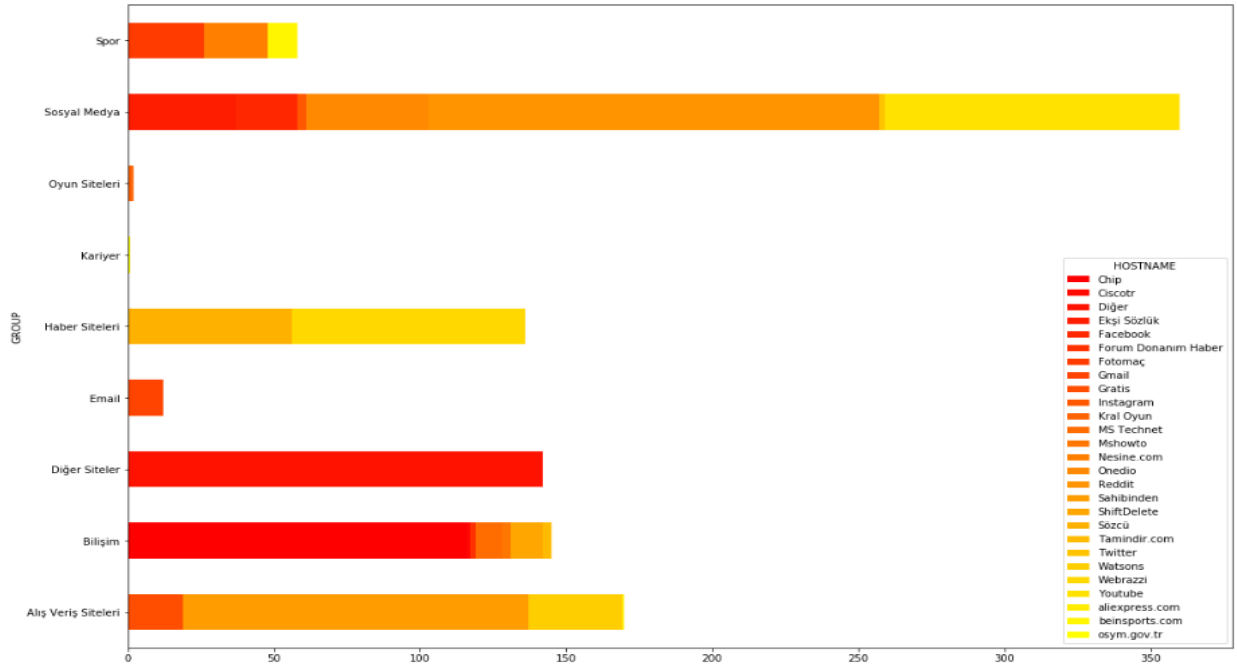
```
<matplotlib.axes._subplots.AxesSubplot at 0x7d4cdccc0>
```



Şekil 4.22: Ders notu 50'nin altındaki kadınların gittiği web sitelerin dağılımı

```
datav2.plot(kind='barh', stacked=True, figsize=[18,13], colormap='autumn')
```

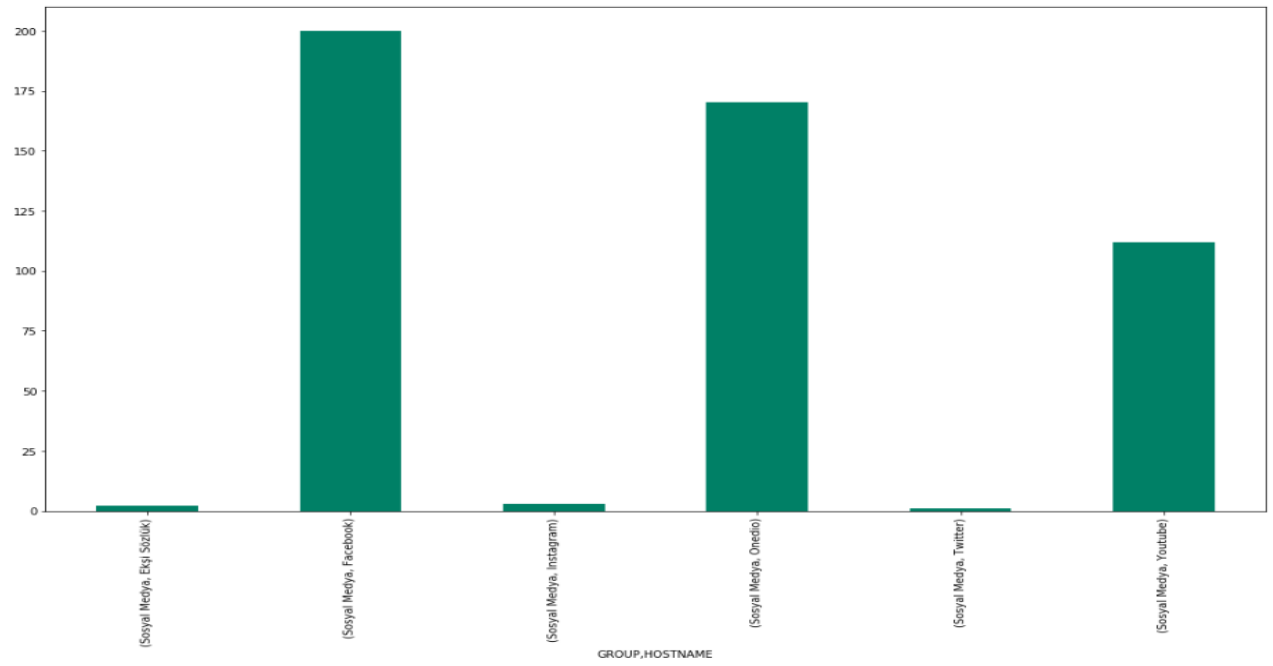
```
<matplotlib.axes._subplots.AxesSubplot at 0x7d1b4cbe0>
```



Şekil 4.23: Ders notu 50'nin altındaki erkeklerin gittiği web sitelerin dağılımı

```
datav2 = (data[(data['GEN'] == 'F') & (data['GROUP'] == 'Sosyal Medya') & (data['ORT'] < 50.00)]
          .groupby(['GROUP','HOSTNAME'])
         .size())
datav2
datav2.plot(kind='bar', stacked=True, figsize=[18,10], colormap='summer')
```

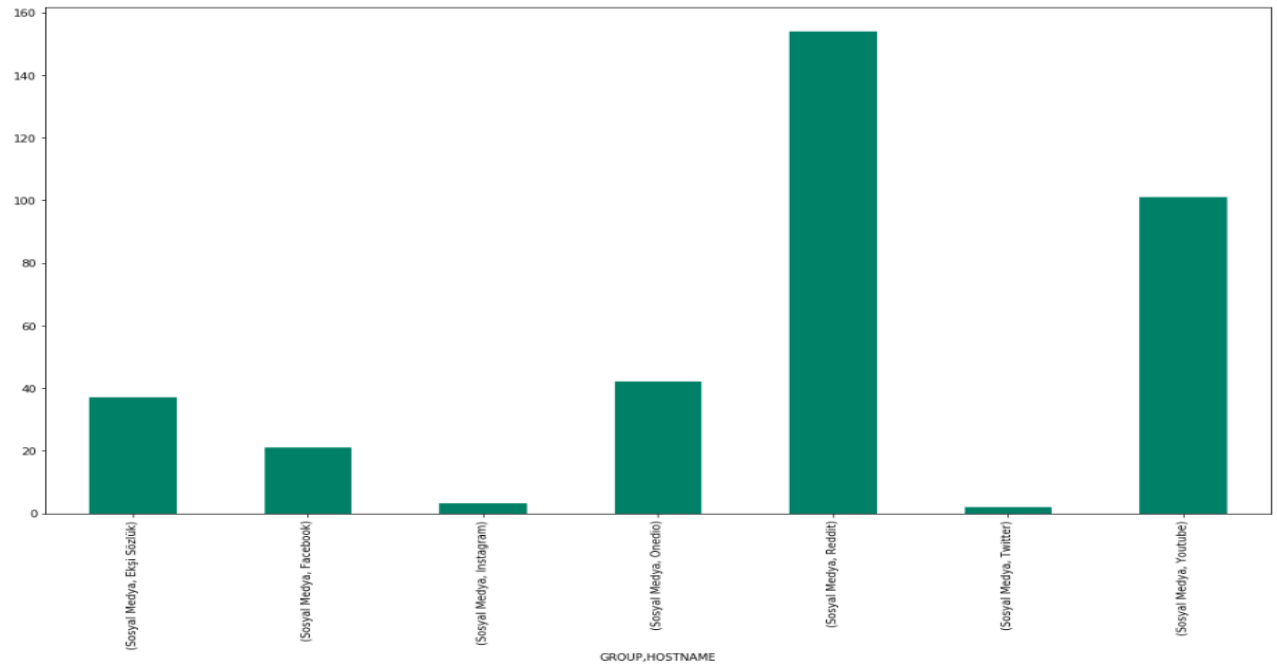
```
<matplotlib.axes._subplots.AxesSubplot at 0x7ce4bc3c8>
```



Şekil 4.24: Ders notu 50'nin altındaki kadınların sosyal medya dağılımı

```
datav2 = (data[(data['GEN'] == 'M') & (data['GROUP'] == 'Sosyal Medya') & (data['ORT'] < 50.00)]
          .groupby(['GROUP','HOSTNAME'])
         .size())
datav2
datav2.plot(kind='bar', stacked=True, figsize=[18,10], colormap='summer')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ce5221d0>
```



Şekil 4.25: Ders notu 50'nin altındaki erkeklerin sosyal medya dağılımı

Şekil 4.26'da 50-59 arasında ders notu alan kadın öğrencilerin kod bloğu ve matris yapısı görünmektedir. Şekil 4.27'de 50-59 arasında ders notu alan erkek öğrencilerin kod bloğu ve matris yapısı görünmektedir. Bu matris yapısı ile 50-59 arasında ders notu alan öğrencilerin gittikleri web sitelerin ve bu değerlere bağlı olarak sosyal medya içerikli grafiklerin çıkarılmasına referans olmaktadır.

```
datav2 = (data[(data['GEN'] == 'F') & (data['ORT'] < 60.00) & (data['ORT'] >= 50.00)].groupby(['GROUP','HOSTNAME'])
          .size().unstack()
          .reset_index().fillna(0)
          .set_index('GROUP'))
datav2
```

HOSTNAME	Chip	Ciscotr	Diğer	Ensonhaber	Facebook	Forum Donanım Haber	Gmail	Instagram	Kral Oyun	MS Technet	Mshowto	Onedio	ShiftDelete	Trendyol	Twitter	Wats
GROUP																
Alış Veriş Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	80.0	0.0	
Bilişim	8.0	3.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0	31.0	11.0	0.0	23.0	0.0	0.0	
Diğer Siteler	0.0	0.0	53.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
Email	0.0	0.0	0.0	0.0	0.0	0.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
Haber Siteleri	0.0	0.0	0.0	10.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
Kariyer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
Oyun Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	
Sosyal Medya	0.0	0.0	0.0	0.0	206.0	0.0	0.0	40.0	0.0	0.0	0.0	39.0	0.0	0.0	25.0	
Spor	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

Şekil 4.26: Ders notu 50-59 arasındaki kadın öğrencilerin matris görünümü

```
datav2 = (data[(data['GEN'] == 'M') & (data['ORT'] < 60.00) & (data['ORT'] >= 50.00)].groupby(['GROUP','HOSTNAME'])
          .size().unstack()
          .reset_index().fillna(0)
          .set_index('GROUP'))
datav2
```

HOSTNAME	Chip	Ciscotr	Diğer	Ekşi Sözlük	Ensonhaber	Facebook	Forum Donanım Haber	Fotomaç	Gmail	Gratis	...	Tamindir.com	Trendyol	Twitter	Watsons	Webrazzi
GROUP																
Alış Veriş Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.0	...	0.0	2.0	0.0	3.0	0.0
Bilişim	266.0	49.0	0.0	0.0	0.0	0.0	301.0	0.0	0.0	0.0	...	8.0	0.0	0.0	0.0	0.0
Diğer Siteler	0.0	0.0	490.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
Email	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	22.0	0.0	...	0.0	0.0	0.0	0.0	0.0
Haber Siteleri	0.0	0.0	0.0	0.0	84.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	121.0
Kariyer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
Oyun Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
Sosyal Medya	0.0	0.0	0.0	8.0	0.0	873.0	0.0	0.0	0.0	0.0	...	0.0	0.0	249.0	0.0	0.0
Spor	0.0	0.0	0.0	0.0	0.0	0.0	0.0	278.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0

9 rows × 33 columns

Şekil 4.27: Ders notu 50-59 arasındaki erkek öğrencilerin matris görünümü

Şekil 4.28’de 50-59 arasında ders notu alan ve geçer olarak kategorize edilen kadın öğrencilerin erişim sağladıkları web siteler yeralmaktadır.Grafiğin sağ alt kısmında renk skalasıyla grupların içinde, ilgili gruba ait web siteler görünmektedir. Kadınların sosyal medyadan sonra sırasıyla alışveriş, bilişim, diğer sitelere eriştikleri görünmektedir. Alışveriş sitelerinde sırayla Trendyol ve Watsons’a erişim sağladıkları görünmektedir.

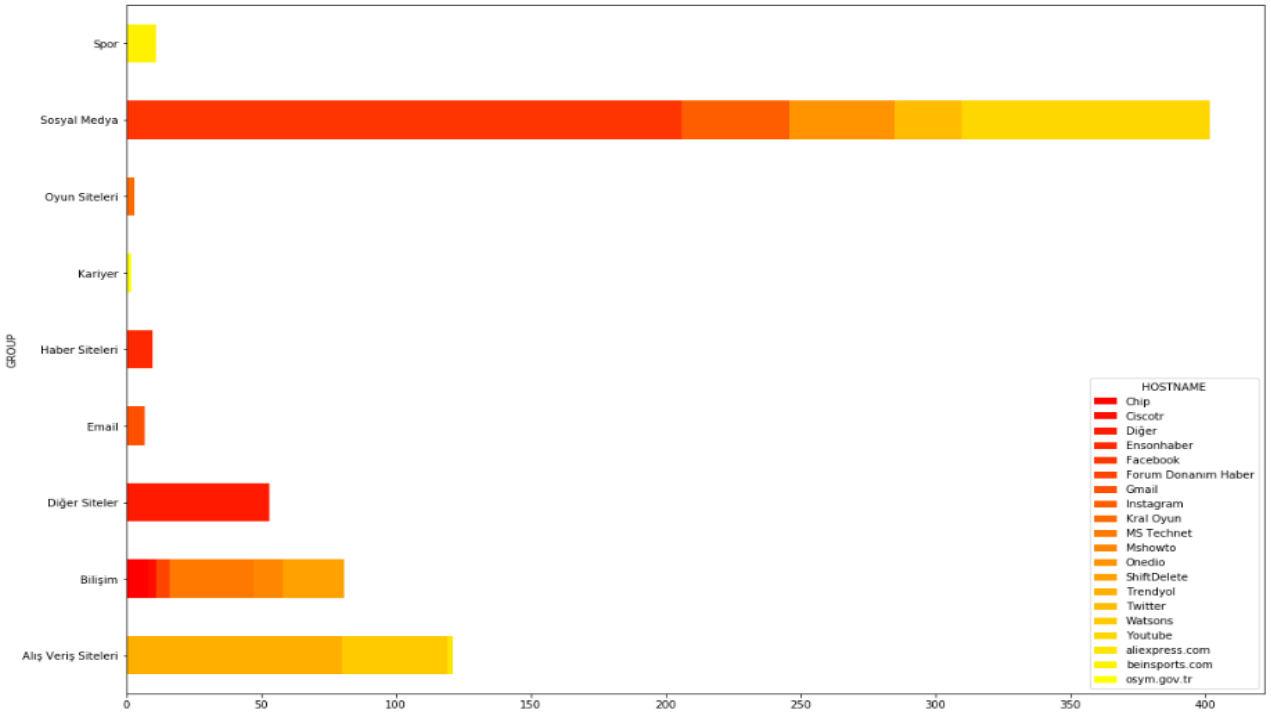
Şekil 4.29’da 50-59 arasında ders notu alan ve geçer olarak kategorize edilen erkek öğrencilerin erişim sağladıkları web siteler yeralmaktadır. Erkek öğrencilerin kadınlara kıyasla sporla ilgili sitelere daha çok eriştikleri görünmektedir.

Şekil 4.30’da 50-59 arasında ders notu alan kadın öğrencilerin sosyal medya içerikli sitelere erişimleri yeralmaktadır. Şeklin üst kısmında sosyal medya içeriklerinin filtrelenmesini sağlayan kod bloğu yeralmaktadır. Kadın öğrencilerin en çok erişim sağladıkları sosyal medya sitelerinin sırasıyla facebook ve youtube olarak görünmektedir.

Şekil 4.31’de 50-59 arasında ders notu alan erkek öğrencilerin sosyal medya içerikli sitelere erişimleri yeralmaktadır. Şeklin üst kısmında sosyal medya içeriklerinin filtrelenmesini sağlayan kod bloğu yeralmaktadır. Erkek öğrencilerin en çok erişim sağladıkları sosyal medya sitelerine sırasıyla youtube ve facebook olduğu görünmektedir.

```
datav2.plot(kind='barh', stacked=True, figsize=[18,13], colormap='autumn')
```

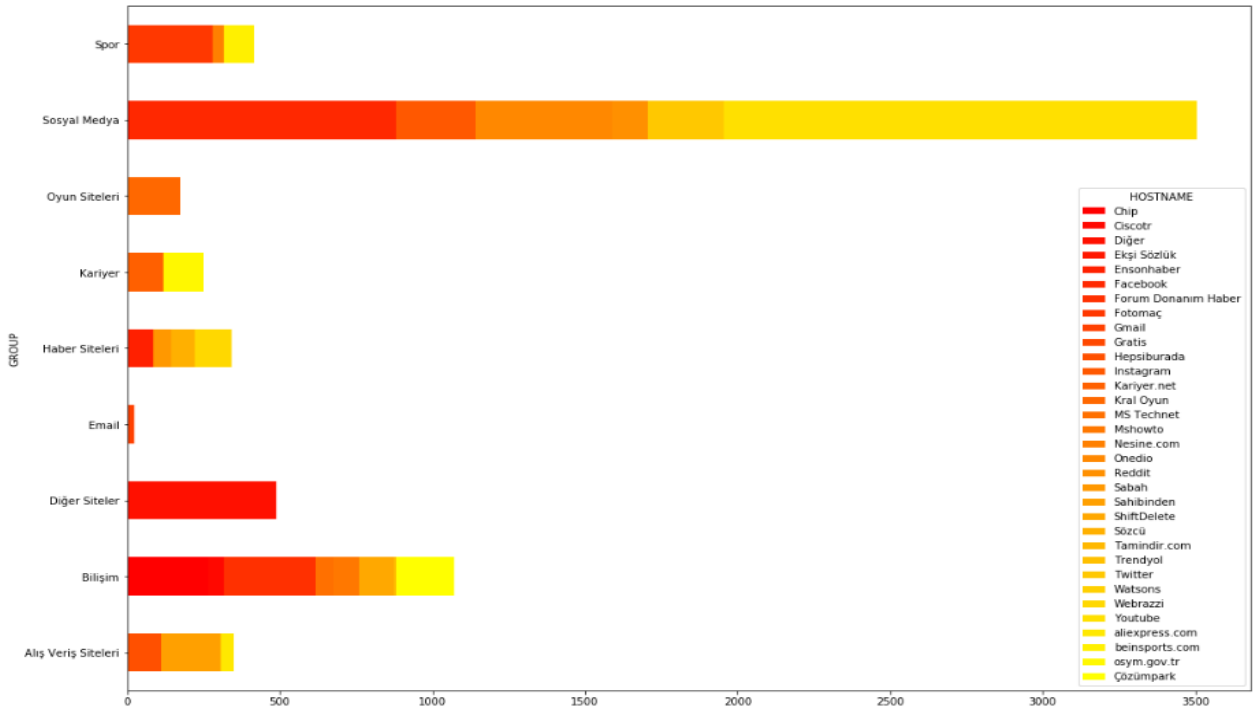
```
<matplotlib.axes._subplots.AxesSubplot at 0x7d1ade5c0>
```



Şekil 4.28: Ders notu 50-59 arasındaki kadınların gittiği web sitelerin dağılımı

```
datav2.plot(kind='barh', stacked=True, figsize=[18,13], colormap='autumn')
```

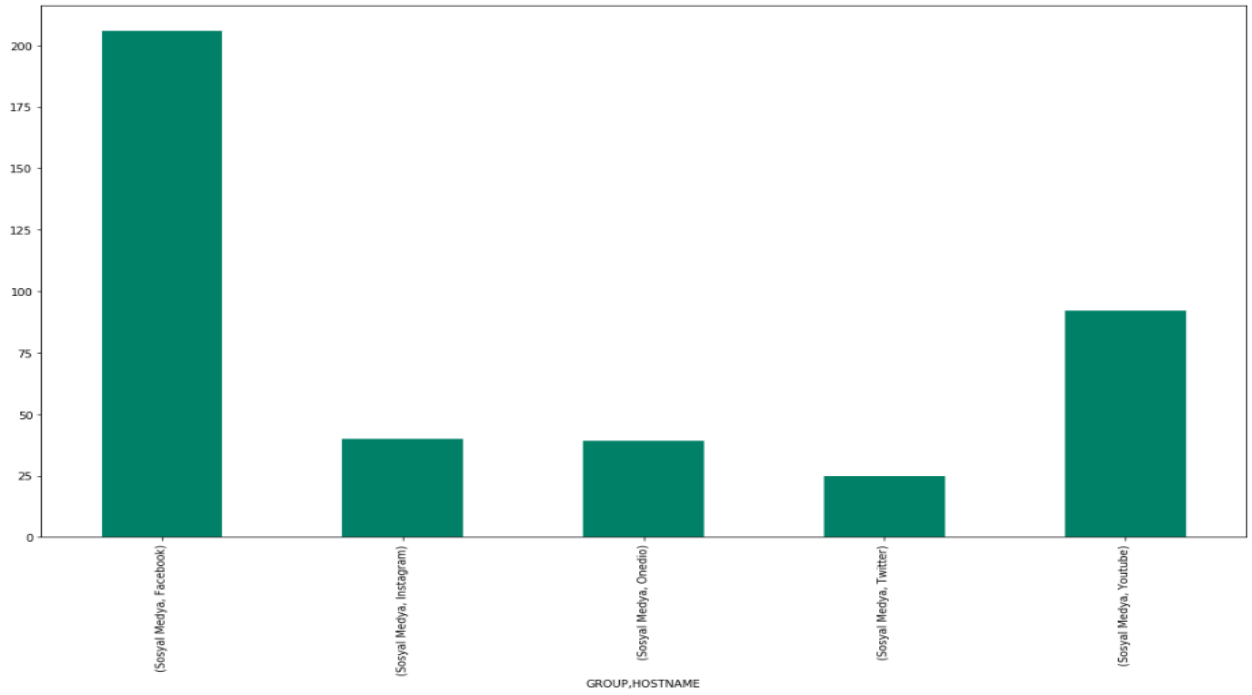
```
<matplotlib.axes._subplots.AxesSubplot at 0x7d3eae4e0>
```



Şekil 4.29: Ders notu 50-59 arasındaki erkeklerin gittiği web sitelerin dağılımı

```
datav2 = (data[(data['GEN'] == 'F') & (data['GROUP'] == 'Sosyal Medya') & (data['ORT'] < 60.00) & (data['ORT'] >= 50.00)]
          .groupby(['GROUP','HOSTNAME'])
         .size())
datav2
datav2.plot(kind='bar', stacked=True, figsize=[18,10], colormap='summer')
```

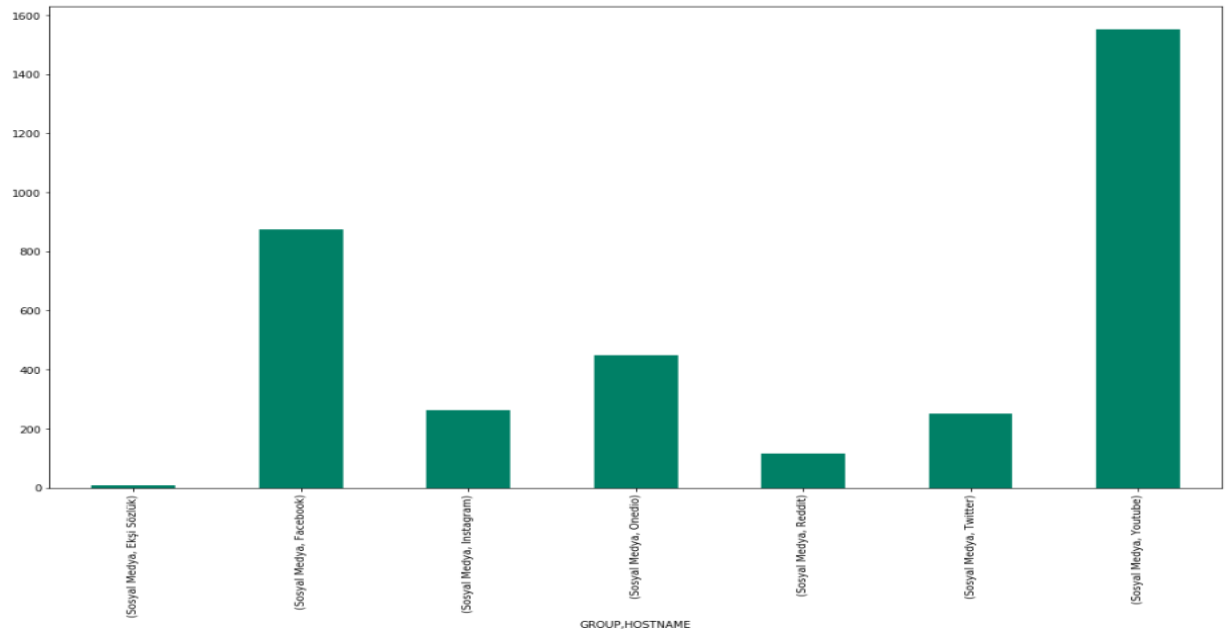
```
<matplotlib.axes._subplots.AxesSubplot at 0x7cdddb1d0>
```



Şekil 4.30: Ders notu 50-59 arasındaki kadınların sosyal medya dağılımı

```
datav2 = (data[(data['GEN'] == 'M') & (data['GROUP'] == 'Sosyal Medya') & (data['ORT'] < 60.00) & (data['ORT'] >= 50.00)]
          .groupby(['GROUP','HOSTNAME'])
         .size())
datav2
datav2.plot(kind='bar', stacked=True, figsize=[18,10], colormap='summer')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7cde47470>
```



Şekil 4.31: Ders notu 50-59 arasındaki erkeklerin sosyal medya dağılımı

Şekil 4.32'de 60-79 arasında ders notu alan kadın öğrencilerin kod bloğu ve matris yapısı görünmektedir. Şekil 4.33'de 60-79 arasında ders notu alan erkek öğrencilerin kod bloğu ve matris yapısı görünmektedir. Bu matris yapısı ile 60-79 arasında ders notu alan öğrencilerin gittikleri web sitelerin ve bu değerlere bağlı olarak sosyal medya içerikli grafiklerin çıkarılmasına referans olmaktadır.

```
datav2 = (data[(data['GEN'] == 'F') & (data['ORT'] < 80.00) & (data['ORT'] >= 60.00)].groupby(['GROUP','HOSTNAME'])
          .size().unstack()
          .reset_index().fillna(0)
          .set_index('GROUP'))
datav2
```

HOSTNAME	Chip	Ciscotr	Diğer	Ekşi Sözlük	Ensonhaber	Facebook	Forum Donanım Haber	Fotomaç	Gmail	Gratis	...	ShiftDelete	Tamindir.com	Trendyol	Twitter	Watson
GROUP																
Alış Veriş Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	163.0	...	0.0	0.0	73.0	0.0	80.
Bilişim	41.0	13.0	0.0	0.0	0.0	0.0	72.0	0.0	0.0	0.0	...	124.0	20.0	0.0	0.0	0.
Diğer Siteler	0.0	0.0	248.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.
Email	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	38.0	0.0	...	0.0	0.0	0.0	0.0	0.
Haber Siteleri	0.0	0.0	0.0	0.0	24.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.
Kariyer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.
Oyun Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.
Sosyal Medya	0.0	0.0	0.0	8.0	0.0	425.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	15.0	0.
Spor	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.

9 rows × 29 columns

Şekil 4.32: Ders notu 60-79 arasındaki kadın öğrencilerin matris görünümü

```
datav2 = (data[(data['GEN'] == 'M') & (data['ORT'] < 80.00) & (data['ORT'] >= 60.00)].groupby(['GROUP','HOSTNAME'])
          .size().unstack()
          .reset_index().fillna(0)
          .set_index('GROUP'))
datav2
```

HOSTNAME	Chip	Ciscotr	Diğer	Ekşi Sözlük	Ensonhaber	Facebook	Forum Donanım Haber	Fotomaç	Gmail	Gratis	...	Tamindir.com	Trendyol	Twitter	Watsons	Webrazz
GROUP																
Alış Veriş Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	49.0	...	0.0	84.0	0.0	101.0	0.(
Bilişim	350.0	259.0	0.0	0.0	0.0	0.0	232.0	0.0	0.0	0.0	...	18.0	0.0	0.0	0.0	0.(
Diğer Siteler	0.0	0.0	1143.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.(
Email	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	137.0	0.0	...	0.0	0.0	0.0	0.0	0.(
Haber Siteleri	0.0	0.0	0.0	0.0	310.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	259.(
Kariyer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.(
Oyun Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.(
Sosyal Medya	0.0	0.0	0.0	259.0	0.0	1701.0	0.0	0.0	0.0	0.0	...	0.0	0.0	303.0	0.0	0.(
Spor	0.0	0.0	0.0	0.0	0.0	0.0	0.0	435.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.(

9 rows × 33 columns

Şekil 4.33: Ders notu 60-79 arasındaki erkek öğrencilerin matris görünümü

Şekil 4.34'de 60-79 arasında ders notu alan ve orta olarak kategorize edilen kadın öğrencilerin erişim sağladıkları web siteler yeralmaktadır. Grafiğin sağ alt kısmında renk skalasıyla grupların içinde, ilgili gruba ait web siteler görünmektedir. Grafiğin üst kısmında kod bloğu yeralmaktadır. Kadınların sosyal medyadan sonra sırasıyla alışveriş, bilişim, diğer sitelere eriştikleri görünmektedir.

Şekil 4.35'de 60-79 arasında ders notu alan ve orta olarak kategorize edilen erkek öğrencilerin erişim sağladıkları web siteler yeralmaktadır. Erkek öğrencilerinde kadın öğrencilerin erişim sağladıkları sitelere gittikleri görülmektedir. Ayrıca erkeklerin kadınlara kıyasla haber siteleriyle ilgili sitelere daha çok eriştikleri görünmektedir.

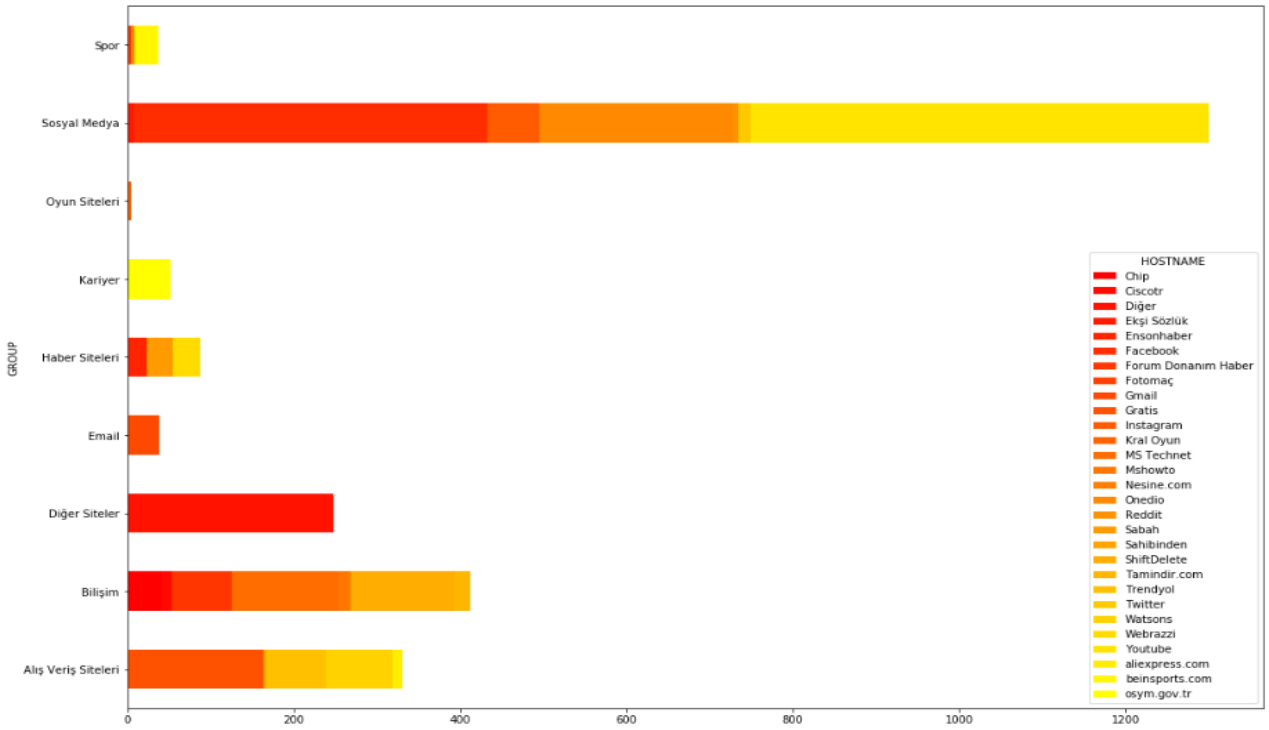
Şekil 4.36'da 60-79 arasında ders notu alan kadın öğrencilerin sosyal medya içerikli sitelere erişimleri yeralmaktadır. Şeklin üst kısmında sosyal medya içeriklerinin filtrelenmesini sağlayan kod bloğu yeralmaktadır. Kadın öğrencilerin en çok erişim sağladıkları sosyal medya sitelerinin sırasıyla facebook,youtube ve onedio olarak görünmektedir.

Şekil 4.37'de 60-79 arasında ders notu alan erkek öğrencilerin sosyal medya içerikli sitelere erişimleri yeralmaktadır. Şeklin üst kısmında sosyal medya içeriklerinin filtrelenmesini sağlayan kod bloğu yeralmaktadır. Erkek öğrencilerinde kadın öğrenciler gibi aynı sitelere erişim sağladıkları tespit edilmiştir.

Her grubun içeriği sağ alt kısımdaki renk skalasından kontrol edilip erişim sağlanılan siteler grup içerisinden tespit edilebilmektedir.

```
datav2.plot(kind='barh', stacked=True, figsize=[18,13], colormap='autumn')
```

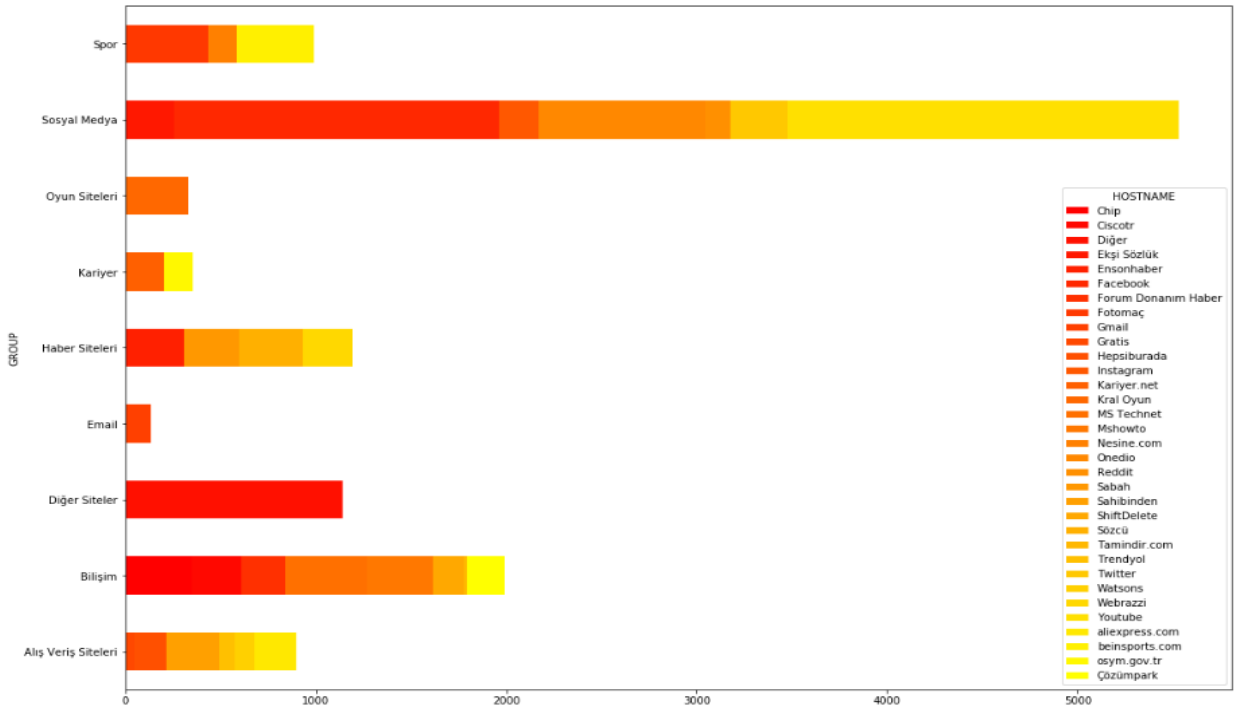
```
<matplotlib.axes._subplots.AxesSubplot at 0x7d0602cf8>
```



Şekil 4.34: Ders notu 60-79 arasındaki kadınların gittiği web sitelerin dağılımı

```
datav2.plot(kind='barh', stacked=True, figsize=[18,13], colormap='autumn')
```

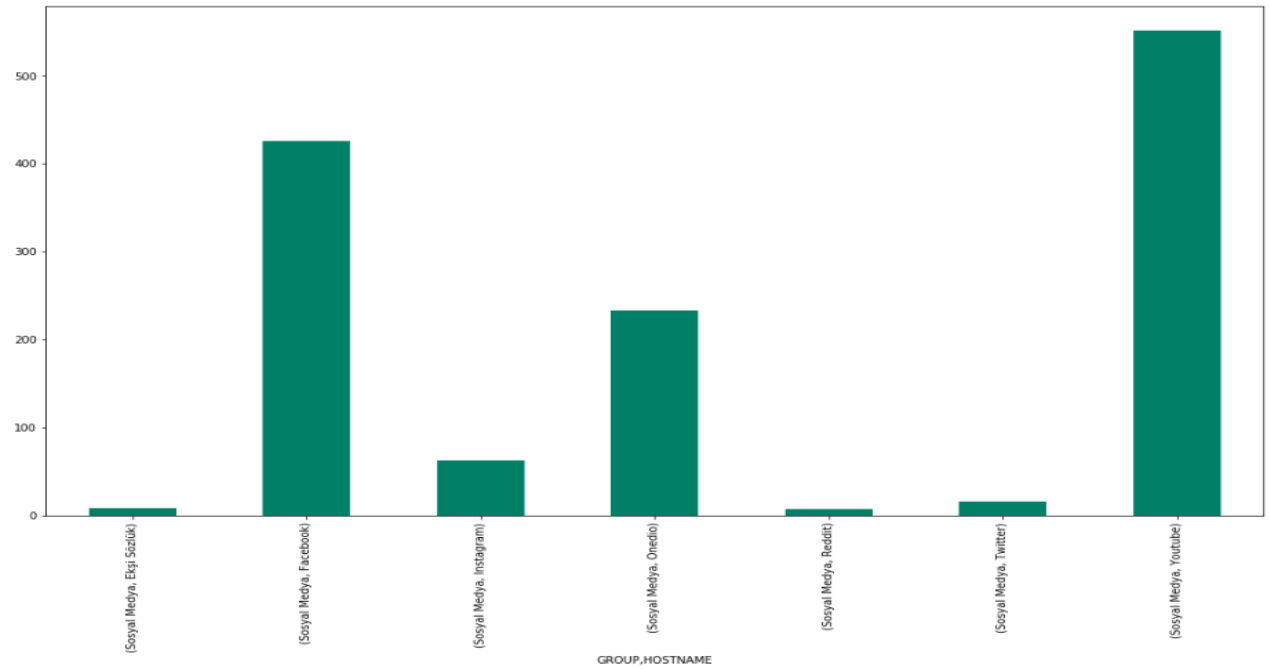
```
<matplotlib.axes._subplots.AxesSubplot at 0x7d138df28>
```



Şekil 4.35: Ders notu 60-79 arasındaki erkeklerin gittiği web sitelerin dağılımı

```
datav2 = (data[(data['GEN'] == 'F') & (data['GROUP'] == 'Sosyal Medya') & (data['ORT'] < 80.00) & (data['ORT'] >= 60.00)]
          .groupby(['GROUP','HOSTNAME'])
         .size())
datav2
datav2.plot(kind='bar', stacked=True, figsize=[18,10], colormap='summer')
```

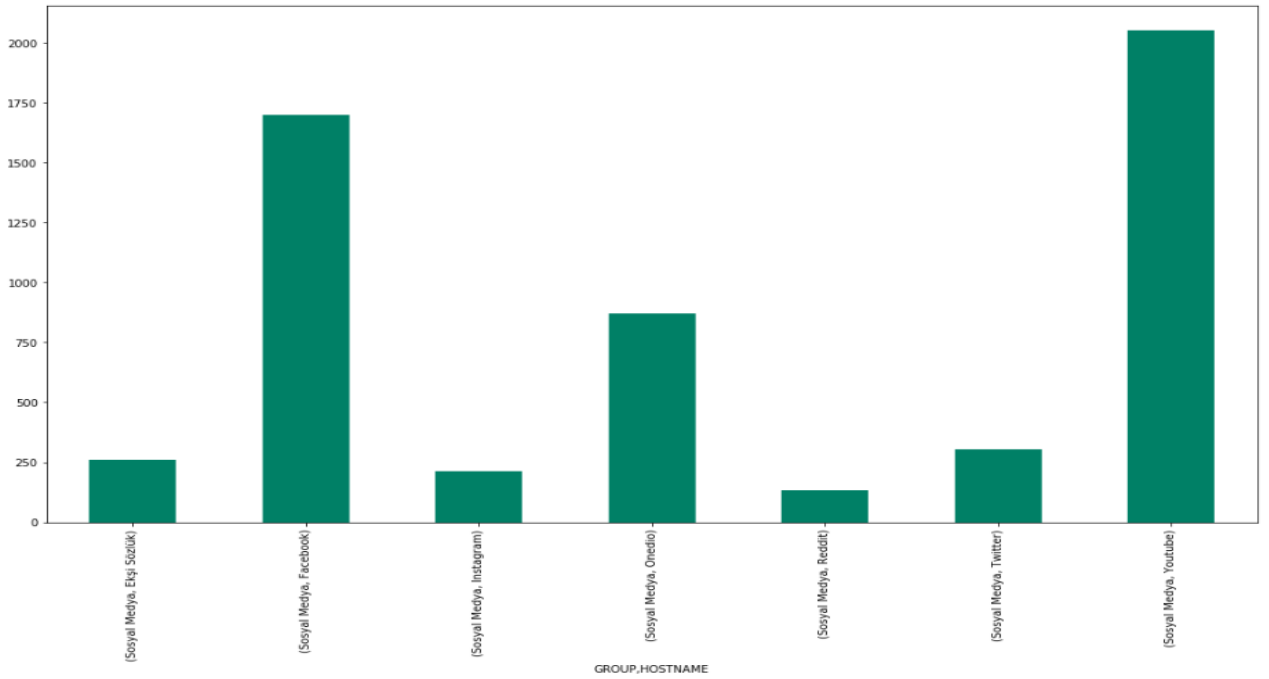
```
<matplotlib.axes._subplots.AxesSubplot at 0x7c418d240>
```



Şekil 4.36: Ders notu 60-79 arasındaki kadınların sosyal medya dağılımı

```
datav2 = (data[(data['GEN'] == 'M') & (data['GROUP'] == 'Sosyal Medya') & (data['ORT'] < 80.00) & (data['ORT'] >= 60.00)]
          .groupby(['GROUP','HOSTNAME'])
         .size())
datav2
datav2.plot(kind='bar', stacked=True, figsize=[18,10], colormap='summer')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7cd721208>
```



Şekil 4.37: Ders notu 60-79 arasındaki erkeklerin sosyal medya dağılımı

Şekil 4.38'da 80-89 arasında ders notu alan kadın öğrencilerin kod bloğu ve matris yapısı görünmektedir. Şekil 4.39'da 80-89 arasında ders notu alan erkek öğrencilerin kod bloğu ve matris yapısı görünmektedir. Bu matris yapısı ile 80-89 arasında ders notu alan öğrencilerin gittikleri web sitelerin ve bu değerlere bağlı olarak sosyal medya içerikli grafiklerin çıkarılmasına referans olmaktadır.

```
datav2 = (data[(data['GEN'] == 'F') & (data['ORT'] < 90.00) & (data['ORT'] >= 80.00)].groupby(['GROUP','HOSTNAME'])
          .size().unstack()
          .reset_index().fillna(0)
          .set_index('GROUP'))
datav2
```

HOSTNAME	Chip	Ciscotr	Diğer	Ekşi Sözlük	Ensonhaber	Facebook	Forum Donanım Haber	Fotomaç	Gratis	Hepsiburada	...	Sabah	Sahibinden	ShiftDelete	Trendyol	Tw
GROUP																
Alış Veriş Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	109.0	55.0	...	0.0	2.0	0.0	132.0	
Bilişim	39.0	81.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0	...	0.0	0.0	103.0	0.0	
Diğer Siteler	0.0	0.0	167.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
Haber Siteleri	0.0	0.0	0.0	0.0	123.0	0.0	0.0	0.0	0.0	0.0	...	12.0	0.0	0.0	0.0	
Kariyer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
Oyun Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
Sosyal Medya	0.0	0.0	0.0	3.0	0.0	442.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
Spor	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	

8 rows × 28 columns

Şekil 4.38: Ders notu 80-89 arasındaki kadın öğrencilerin matris görünümü

```
datav2 = (data[(data['GEN'] == 'M') & (data['ORT'] < 90.00) & (data['ORT'] >= 80.00)].groupby(['GROUP','HOSTNAME'])
          .size().unstack()
          .reset_index().fillna(0)
          .set_index('GROUP'))
datav2
```

HOSTNAME	Chip	Ciscotr	Diğer	Ekşi Sözlük	Ensonhaber	Facebook	Forum Donanım Haber	Fotomaç	Gmail	Gratis	...	Sözcü	Tamindir.com	Trendyol	Twitter	Webrazzi
GROUP																
Alış Veriş Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	11.0	...	0.0	0.0	25.0	0.0	0.0
Bilişim	282.0	17.0	0.0	0.0	0.0	0.0	106.0	0.0	0.0	0.0	...	0.0	7.0	0.0	0.0	0.0
Diğer Siteler	0.0	0.0	338.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
Email	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	33.0	0.0	...	0.0	0.0	0.0	0.0	0.0
Haber Siteleri	0.0	0.0	0.0	0.0	76.0	0.0	0.0	0.0	0.0	0.0	...	22.0	0.0	0.0	0.0	44.0
Kariyer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
Oyun Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
Sosyal Medya	0.0	0.0	0.0	31.0	0.0	374.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	67.0	0.0
Spor	0.0	0.0	0.0	0.0	0.0	0.0	0.0	20.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0

9 rows × 31 columns

Şekil 4.39: Ders notu 80-89 arasındaki erkek öğrencilerin matris görünümü

Şekil 4.40'da 80-89 arasında ders notu alan ve iyi olarak kategorize edilen kadın öğrencilerin erişim sağladıkları web siteler yeralmaktadır. Grafiğin sağ alt kısmında renk skalasıyla grupların içinde, ilgili gruba ait web siteler görünmektedir. Grafiğin üst kısmında kod bloğu yeralmaktadır. Kadınların sosyal medyadan sonra sırasıyla alışveriş, bilişim, haber sitelerine eriştikleri görünmektedir.

Şekil 4.41'de 80-89 arasında ders notu alan ve iyi olarak kategorize edilen erkek öğrencilerin erişim sağladıkları web siteler yeralmaktadır. Erkek öğrencilerin sosyal medya içerikli sitelerden sonra en çok bilişim ve diğer sitelere eriştikleri görünmektedir.

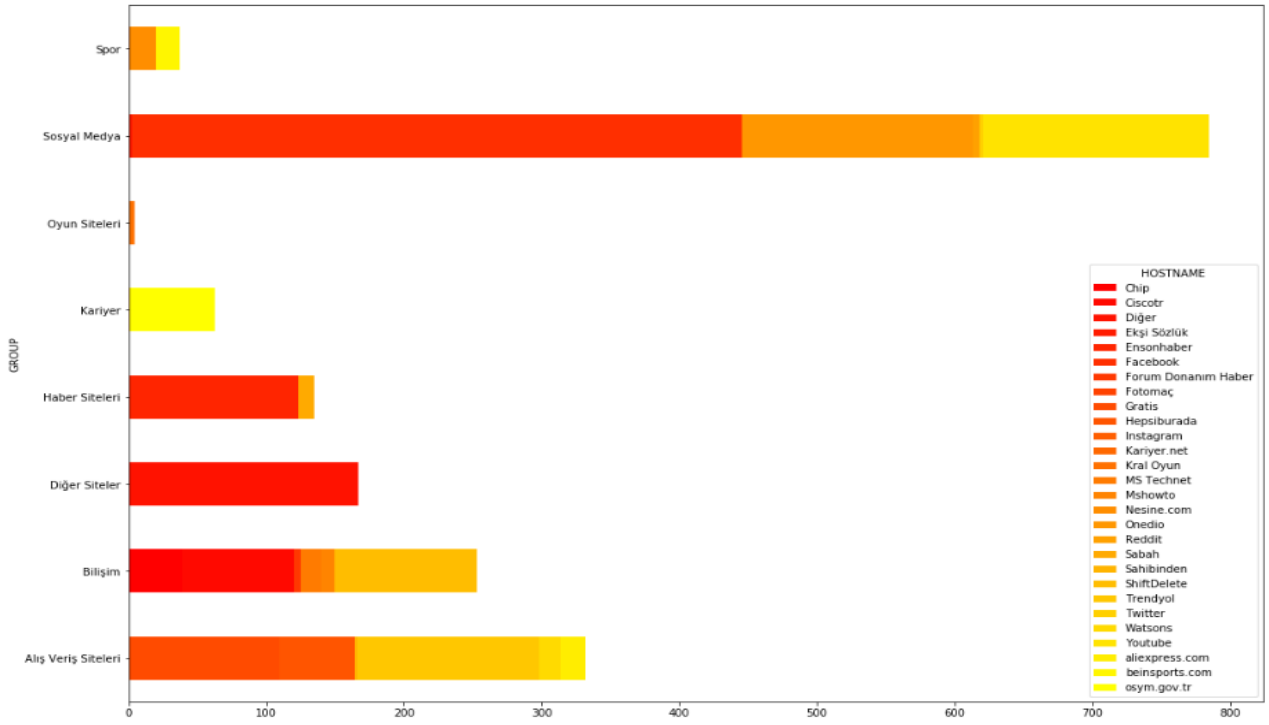
Her grubun içeriği sağ alt kısımdaki renk skalasından kontrol edilip erişim sağlanılan siteler grup içerisinden tespit edilebilmektedir.

Şekil 4.42'de 80-89 arasında ders notu alan kadın öğrencilerin sosyal medya içerikli sitelere erişimleri yeralmaktadır. Şeklin üst kısmında sosyal medya içeriklerinin filtrelenmesini sağlayan kod bloğu yeralmaktadır. Kadın öğrencilerin en çok erişim sağladıkları sosyal medya sitelerinin sırasıyla facebook, youtube ve onedio olarak görünmektedir.

Şekil 4.43'de 80-89 arasında ders notu alan erkek öğrencilerin sosyal medya içerikli sitelere erişimleri yeralmaktadır. Şeklin üst kısmında sosyal medya içeriklerinin filtrelenmesini sağlayan kod bloğu yeralmaktadır. Erkek öğrencilerinde kadın öğrenciler gibi aynı sitelere erişim sağladıkları tespit edilmiştir.

```
datav2.plot(kind='barh', stacked=True, figsize=[18,13], colormap='autumn')
```

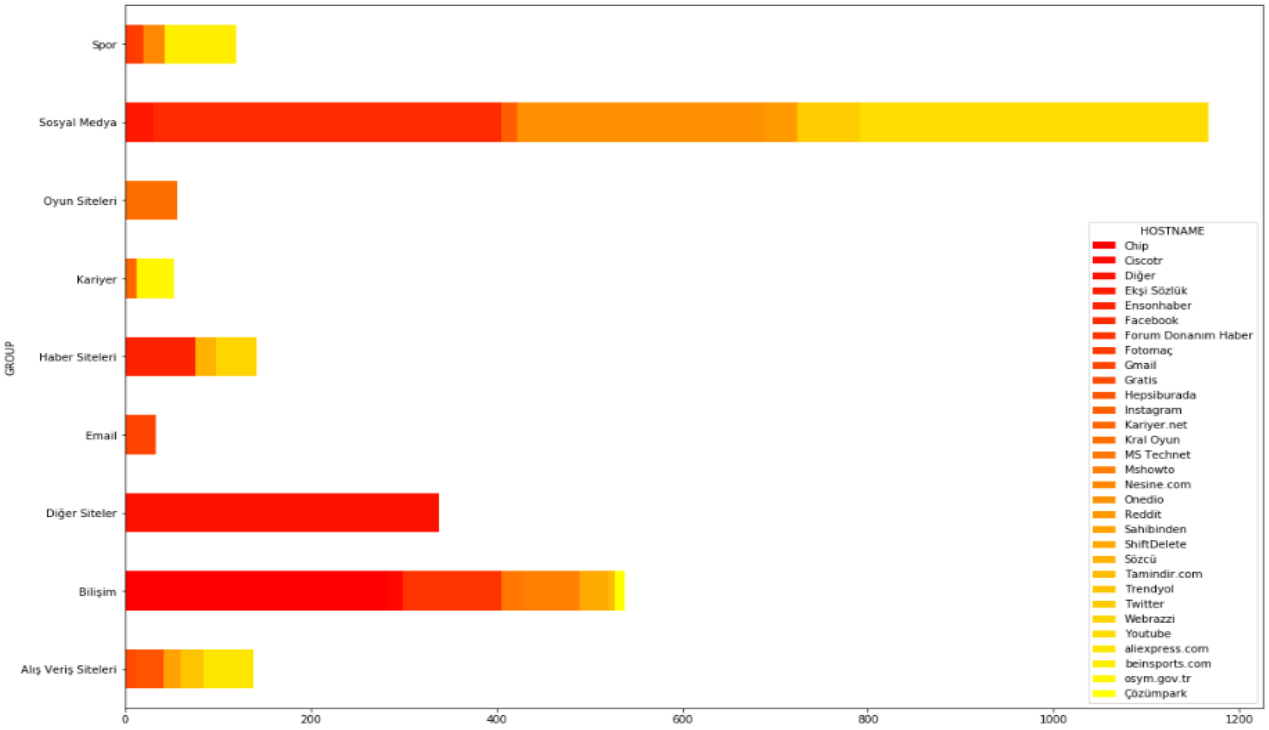
```
<matplotlib.axes._subplots.AxesSubplot at 0x7cfbdaf60>
```



Şekil 4.40: Ders notu 80-89 arasındaki kadınların gittiği web sitelerin dağılımı

```
datav2.plot(kind='barh', stacked=True, figsize=[18,13], colormap='autumn')
```

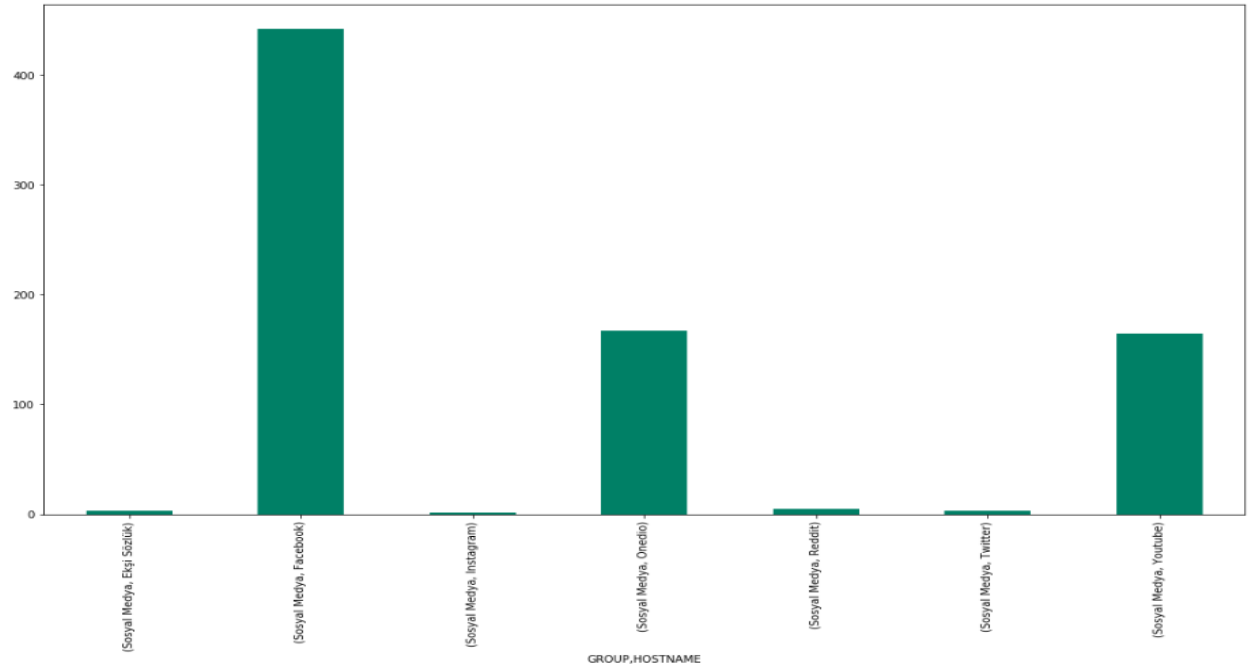
```
<matplotlib.axes._subplots.AxesSubplot at 0x7cfe382e8>
```



Şekil 4.41: Ders notu 80-89 arasındaki erkeklerin gittiği web sitelerin dağılımı

```
datav2 = (data[(data['GEN'] == 'F') & (data['GROUP'] == 'Sosyal Medya') & (data['ORT'] < 90.00) & (data['ORT'] >= 80.00)]
          .groupby(['GROUP','HOSTNAME'])
         .size())
datav2
datav2.plot(kind='bar', stacked=True, figsize=[18,10], colormap='summer')
```

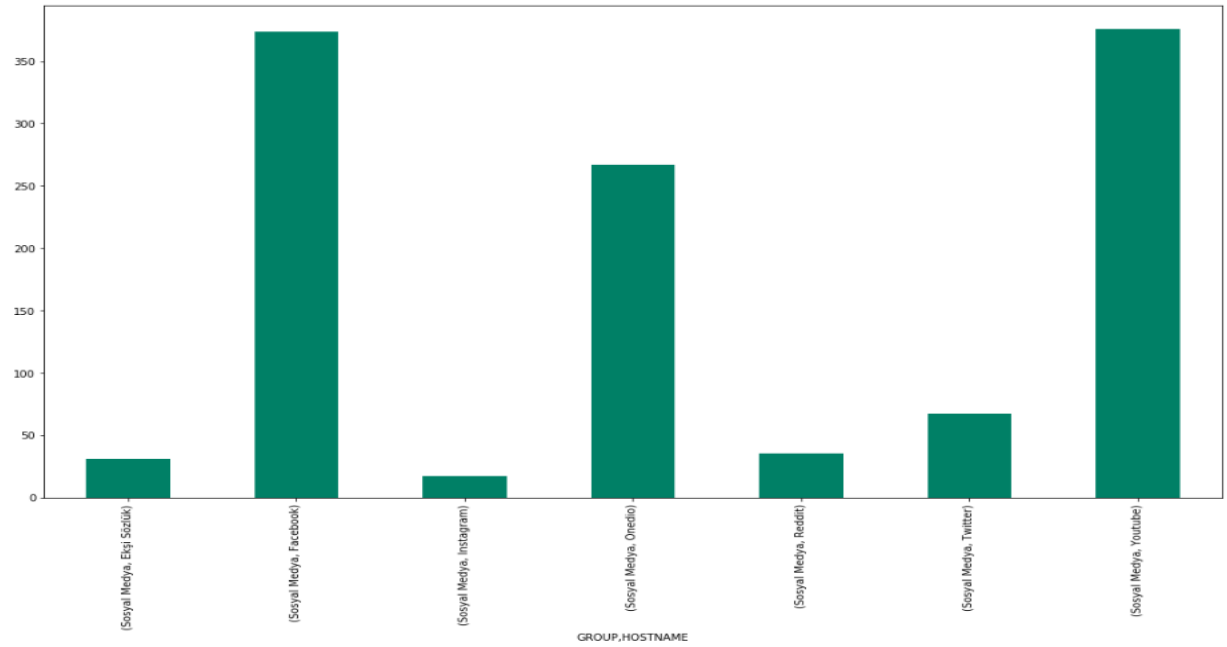
```
<matplotlib.axes._subplots.AxesSubplot at 0x7cc767ba8>
```



Şekil 4.42: Ders notu 80-89 arasındaki kadınların sosyal medya dağılımı

```
datav2 = (data[(data['GEN'] == 'M') & (data['GROUP'] == 'Sosyal Medya') & (data['ORT'] < 90.00) & (data['ORT'] >= 80.00)]
          .groupby(['GROUP','HOSTNAME'])
         .size())
datav2
datav2.plot(kind='bar', stacked=True, figsize=[18,10], colormap='summer')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ccb23400>
```



Şekil 4.43: Ders notu 80-89 arasındaki erkeklerin sosyal medya dağılımı

Şekil 4.44'de 90'dan yüksek ders notu alan kadın öğrencilerin kod bloğu ve matris yapısı görünmektedir. Şekil 4.45'de 90'dan yüksek ders notu alan erkek öğrencilerin kod bloğu ve matris yapısı görünmektedir. Bu matris yapısı ile 90'dan yüksek ders notu alan öğrencilerin gittikleri web sitelerin ve bu değerlere bağlı olarak sosyal medya içerikli grafiklerin çıkarılmasına referans olmaktadır.

```
datav2 = (data[(data['GEN'] == 'F') & (data['ORT'] >= 90.00)].groupby(['GROUP','HOSTNAME'])
          .size().unstack()
          .reset_index().fillna(0)
          .set_index('GROUP'))
datav2
```

HOSTNAME	Chip	Ciscotr	Diğer	Ekşi Sözlük	Ensonhaber	Facebook	Forum Donanım Haber	Gratis	Hepsiburada	Instagram	...	Onedio	Reddit	ShiftDelete	Trendyol	Twitte
GROUP																
Alış Veriş Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	43.0	1.0	0.0	...	0.0	0.0	0.0	11.0	0.
Bilişim	7.0	3.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	...	0.0	0.0	19.0	0.0	0.
Diğer Siteler	0.0	0.0	117.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.
Haber Siteleri	0.0	0.0	0.0	0.0	18.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.
Kariyer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.
Oyun Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.
Sosyal Medya	0.0	0.0	0.0	1.0	0.0	63.0	0.0	0.0	0.0	11.0	...	77.0	6.0	0.0	0.0	76.
Spor	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.

8 rows × 24 columns

Şekil 4.44: Ders notu 90'dan büyük olan kadın öğrencilerin matris görünümü

```
datav2 = (data[(data['GEN'] == 'M') & (data['ORT'] >= 90.00)].groupby(['GROUP','HOSTNAME'])
          .size().unstack()
          .reset_index().fillna(0)
          .set_index('GROUP'))
datav2
```

HOSTNAME	Chip	Ciscotr	Diğer	Ekşi Sözlük	Ensonhaber	Facebook	Forum Donanım Haber	Fotomaç	Instagram	Kral Oyun	...	Sahibinden	ShiftDelete	Trendyol	Twitter	Watsc
GROUP																
Alış Veriş Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	1.0	0.0	
Bilişim	85.0	1.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	...	0.0	4.0	0.0	0.0	
Diğer Siteler	0.0	0.0	19.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
Haber Siteleri	0.0	0.0	0.0	0.0	76.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
Kariyer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
Oyun Siteleri	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	...	0.0	0.0	0.0	0.0	
Sosyal Medya	0.0	0.0	0.0	3.0	0.0	129.0	0.0	0.0	1.0	0.0	...	0.0	0.0	0.0	16.0	
Spor	0.0	0.0	0.0	0.0	0.0	0.0	0.0	11.0	0.0	0.0	...	0.0	0.0	0.0	0.0	

8 rows × 25 columns

Şekil 4.45: Ders notu 90'dan büyük olan erkek öğrencilerin matris görünümü

Şekil 4.46'da 90'dan yüksek ders notu alan ve çok iyi olarak kategorize edilen kadın öğrencilerin erişim sağladıkları web siteler yeralmaktadır. Grafiğin sağ alt kısmında renk skalasıyla grupların içinde, ilgili gruba ait web siteler görünmektedir. Grafiğin üst kısmında kod bloğu yeralmaktadır. Kadınların sosyal medyadan sonra sırasıyla diğer siteler, alışveriş siteleri ve kariyer sitelerine eriştikleri görünmektedir.

Şekil 4.47'de 90'dan yüksek ders notu alan ve çok iyi olarak kategorize edilen erkek öğrencilerin erişim sağladıkları web siteler yeralmaktadır. Erkek öğrencilerin haber siteleri içerikli sitelerden sonra en çok sosyal medya ve bilişim sitelerine eriştikleri görünmektedir.

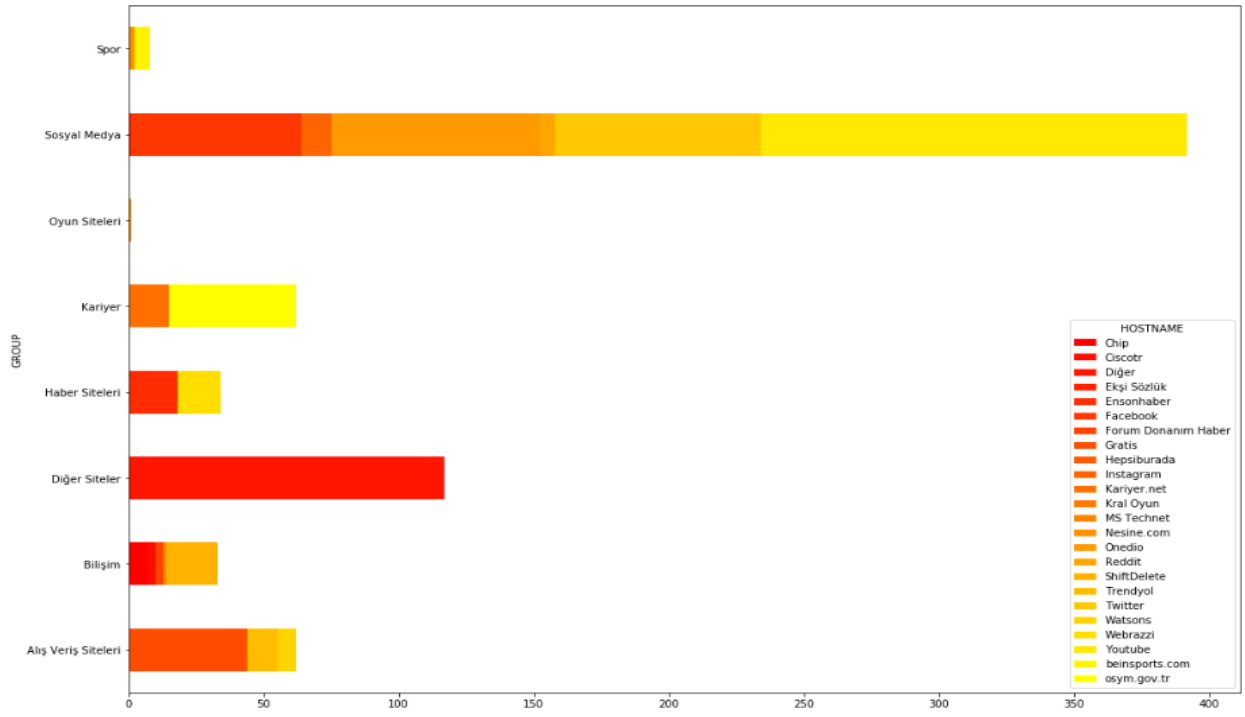
Her grubun içeriği sağ alt kısımdaki renk skalasından kontrol edilip erişim sağlanılan siteler grup içerisinden tespit edilebilmektedir.

Şekil 4.48'de 90'dan yüksek ders notu alan kadın öğrencilerin sosyal medya içerikli sitelere erişimleri yeralmaktadır. Şeklin üst kısmında sosyal medya içeriklerinin filtrelenmesini sağlayan kod bloğu yeralmaktadır. Kadın öğrencilerin en çok erişim sağladıkları sosyal medya sitesinin youtube olduğu tespit edilmiştir. Facebook, Onedio ve Twitter siteleri ise aynı oranlarda erişim sağlandığı görünmüştür.

Şekil 4.49'da 90'dan yüksek ders notu alan erkek öğrencilerin sosyal medya içerikli sitelere erişimleri yeralmaktadır. Şeklin üst kısmında sosyal medya içeriklerinin filtrelenmesini sağlayan kod bloğu yeralmaktadır. Erkek öğrencilerin daha çok facebook'a erişim sağladığı tespit edilmiştir.

```
datav2.plot(kind='barh', stacked=True, figsize=[18,13], colormap='autumn')
```

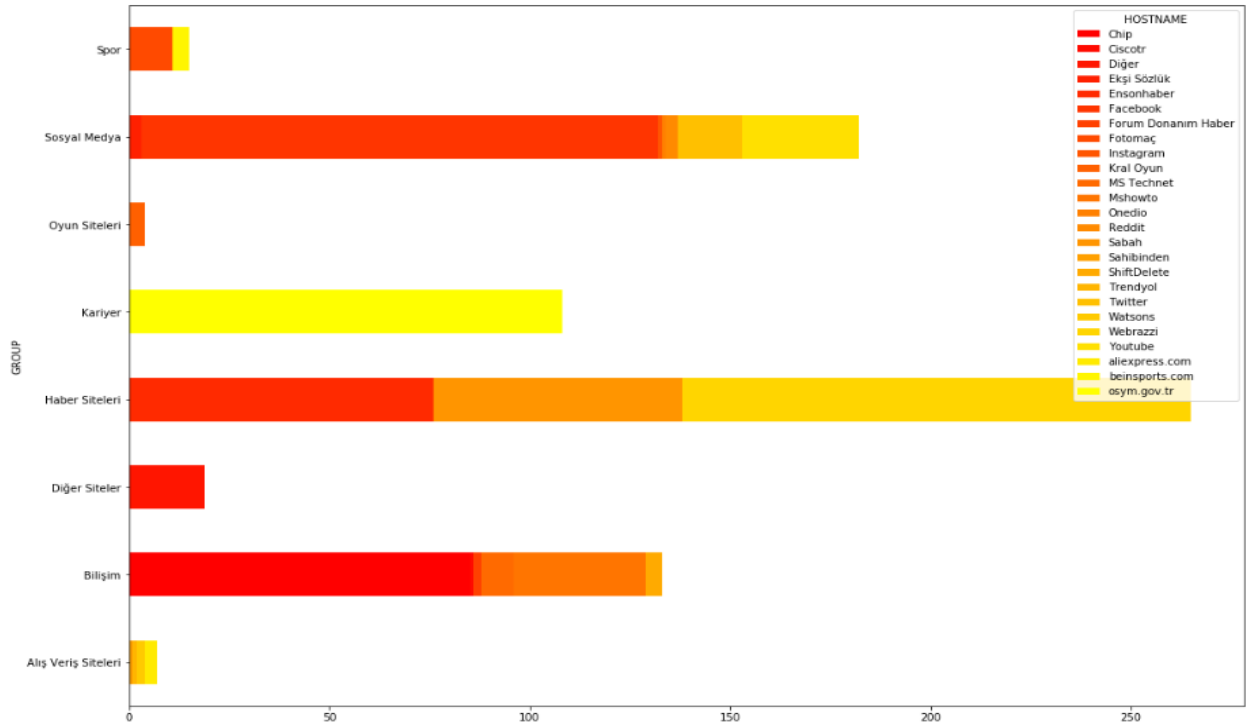
```
<matplotlib.axes._subplots.AxesSubplot at 0x7d61ac7f0>
```



Şekil 4.46: Ders notu 90'dan büyük olan kadınların gittiği web sitelerin dağılımı

```
datav2.plot(kind='barh', stacked=True, figsize=[18,13], colormap='autumn')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7d67e8400>
```



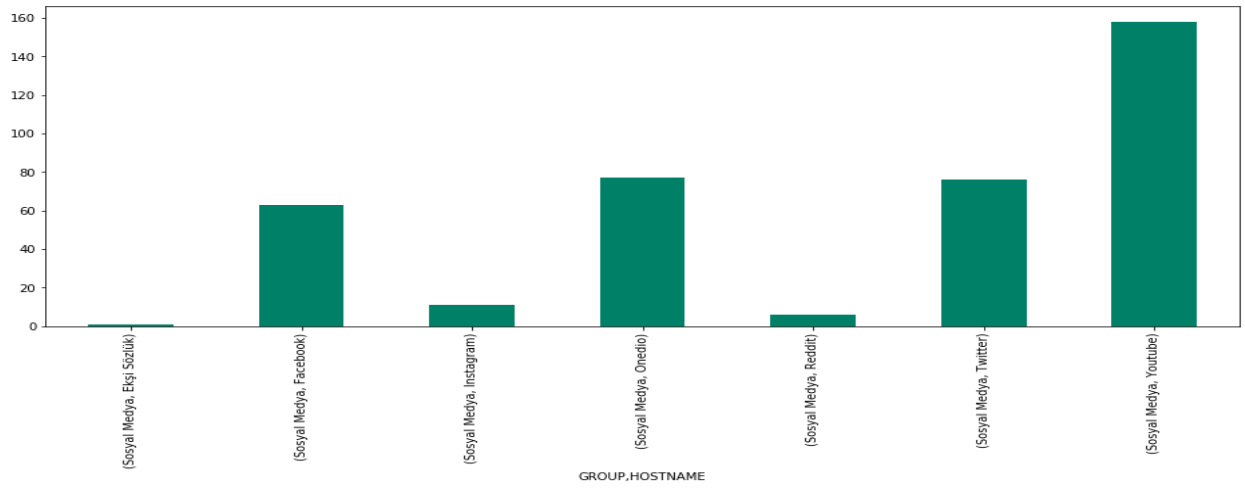
Şekil 4.47: Ders notu 90'dan büyük olan erkeklerin gittiği web sitelerin dağılımı

```
datav2 = (data[(data['GEN'] == 'F') & (data['GROUP'] == 'Sosyal Medya') & (data['ORT'] >= 90.00)]
          .groupby(['GROUP','HOSTNAME'])
        .size())
datav2
```

```
GROUP          HOSTNAME
Sosyal Medya   Ekşi Sözlük      1
               Facebook        63
               Instagram       11
               Onedio          77
               Reddit           6
               Twitter         76
               Youtube        158
dtype: int64
```

```
datav2.plot(kind='bar', stacked=True, figsize=[16,6], colormap='summer')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7c8422198>
```



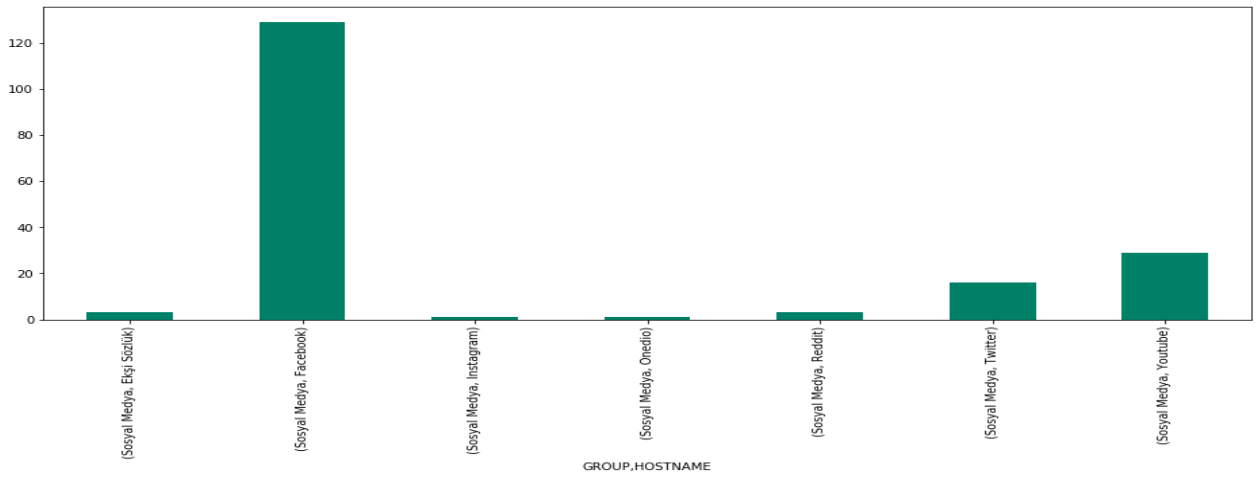
Şekil 4.48: Ders notu 90'dan büyük olan kadınların sosyal medya dağılımı

```
datav2 = (data[(data['GEN'] == 'M') & (data['GROUP'] == 'Sosyal Medya') & (data['ORT'] >= 90.00)]
          .groupby(['GROUP','HOSTNAME'])
        .size())
datav2
```

```
GROUP          HOSTNAME
Sosyal Medya   Ekşi Sözlük      3
               Facebook       129
               Instagram        1
               Onedio           1
               Reddit           3
               Twitter         16
               Youtube         29
dtype: int64
```

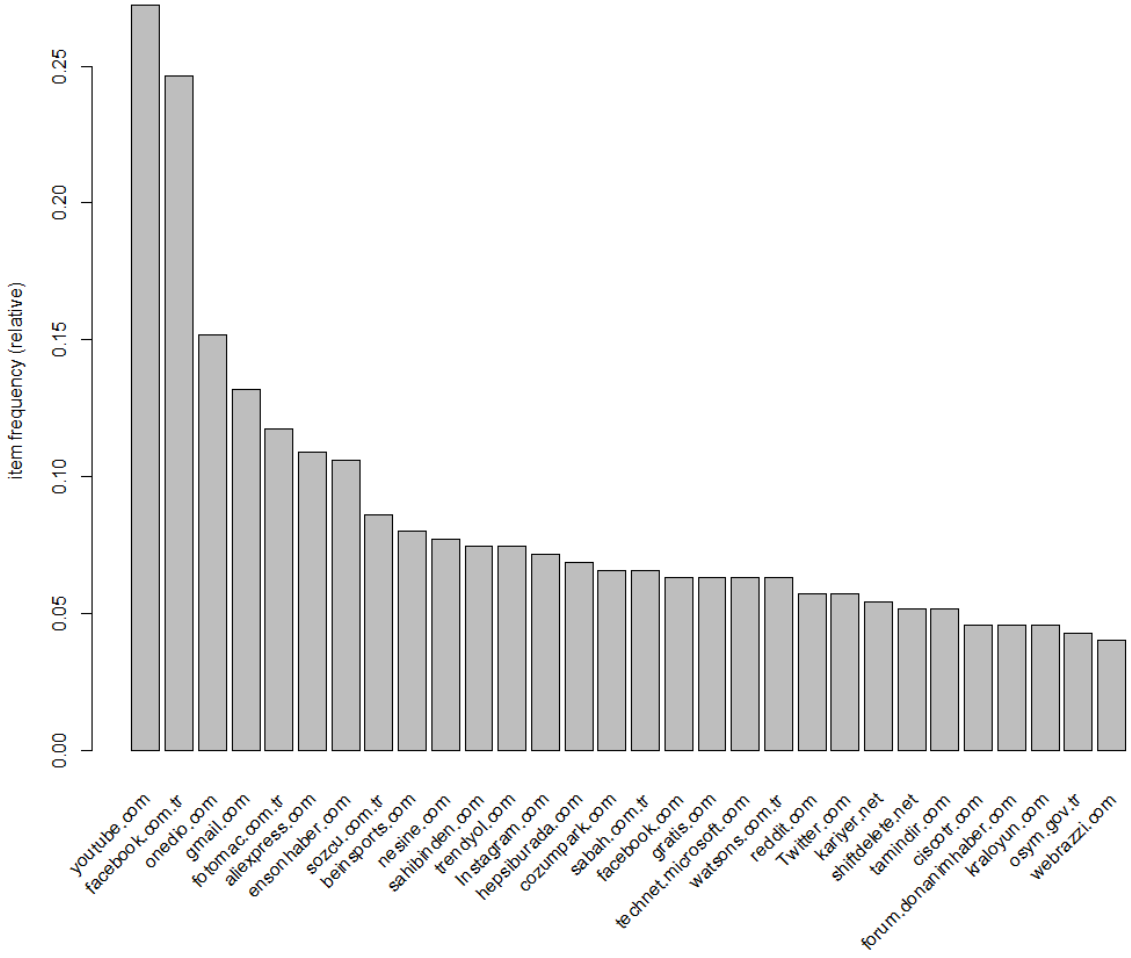
```
datav2.plot(kind='bar', stacked=True, figsize=[16,6], colormap='summer')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7c84945c0>
```



Şekil 4.49: Ders notu 90'dan büyük olan erkeklerin sosyal medya dağılımı

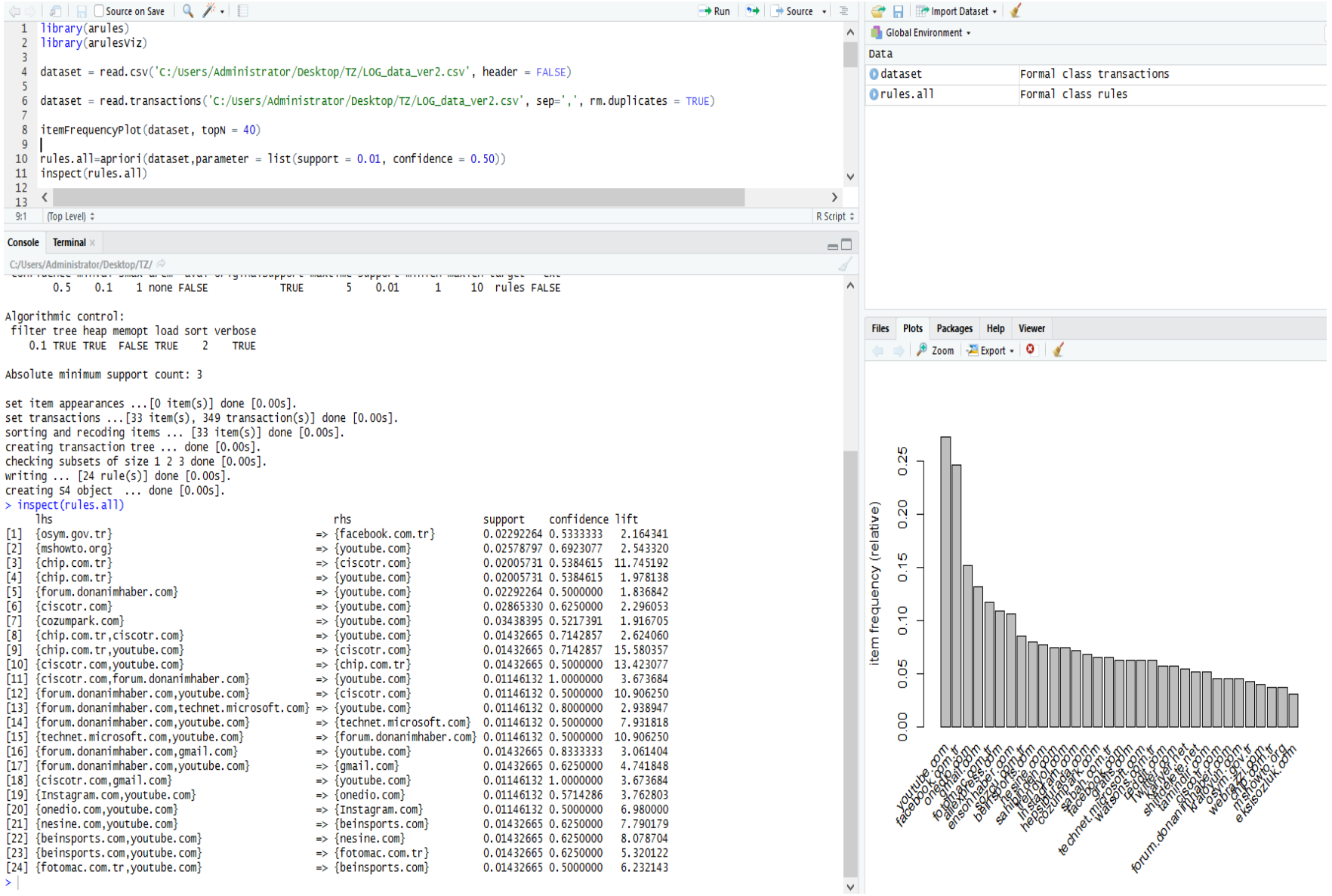
4.4.2 Apriori ve eclat algoritması ile verilerin r studio ortamında değerlendirilmesi



Şekil 4.50: R studio da sık ziyaret edilen web sitelerin grafiği

Bu uygulamada yukarıdaki kurallara göre, destek(**support**) değeri yüzde 1, güven(**confidence**) yüzde 50 olarak belirlenmiştir. Buna göre oluşan kural setinin ilk 24 değeri aşağıda gösterildiği gibidir.

Apriori Algoritması çalıştırıldığında 24 ilişki ortaya çıkmıştır.



Şekil 4.51: R studio da apriori algoritması uygulaması sonucu elde edilen kural setleri

1 numaralı ilişkiyi inceleyecek olursak;

sayı(A) *= sayı("osym.gov.tr") = 15*

sayı(B) *= sayı("facebook.com.tr") = 86*

sayı(A,B) *= sayı("osym.gov.tr" , "facebook.com.tr") = 8*

N *= Tüm kayıtların sayısı = 348*

support(A) *= 15 / 348 → 0,043103*

support(B) *= 86 / 348 → 0,247126*

support(A→B) *= sayı(A,B) / N*

= 8 / 348 → 0,02298851

confidence(A→B) *= sayı(A,B) / sayı(A)*

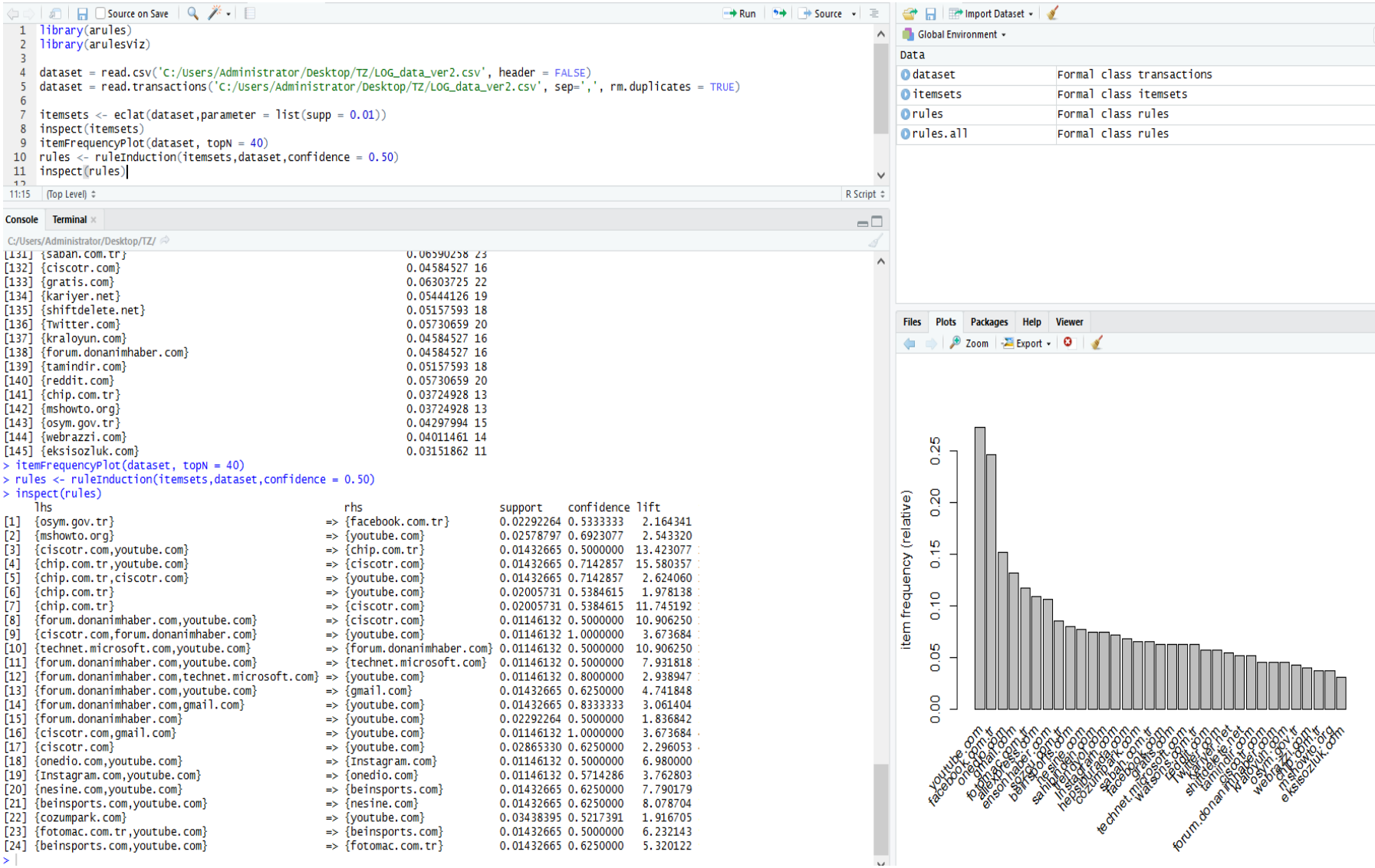
= 8 / 15 → 0,53333

Lift(A→B) *= support(A→B) / support(A) * support(B)*

*= 0,02298851 / (0,043103 * 0,247126) → 2.158140*

Yukarıda Şekil 4.51'den alınan değerler ile hesaplamalar yapılmıştır. Bu hesaplama ile R Studio üzerinde yapılan hesaplamanın doğrulanması amaçlanmıştır. Doğrulamada 1 nolu ilişki gösterilmiştir.

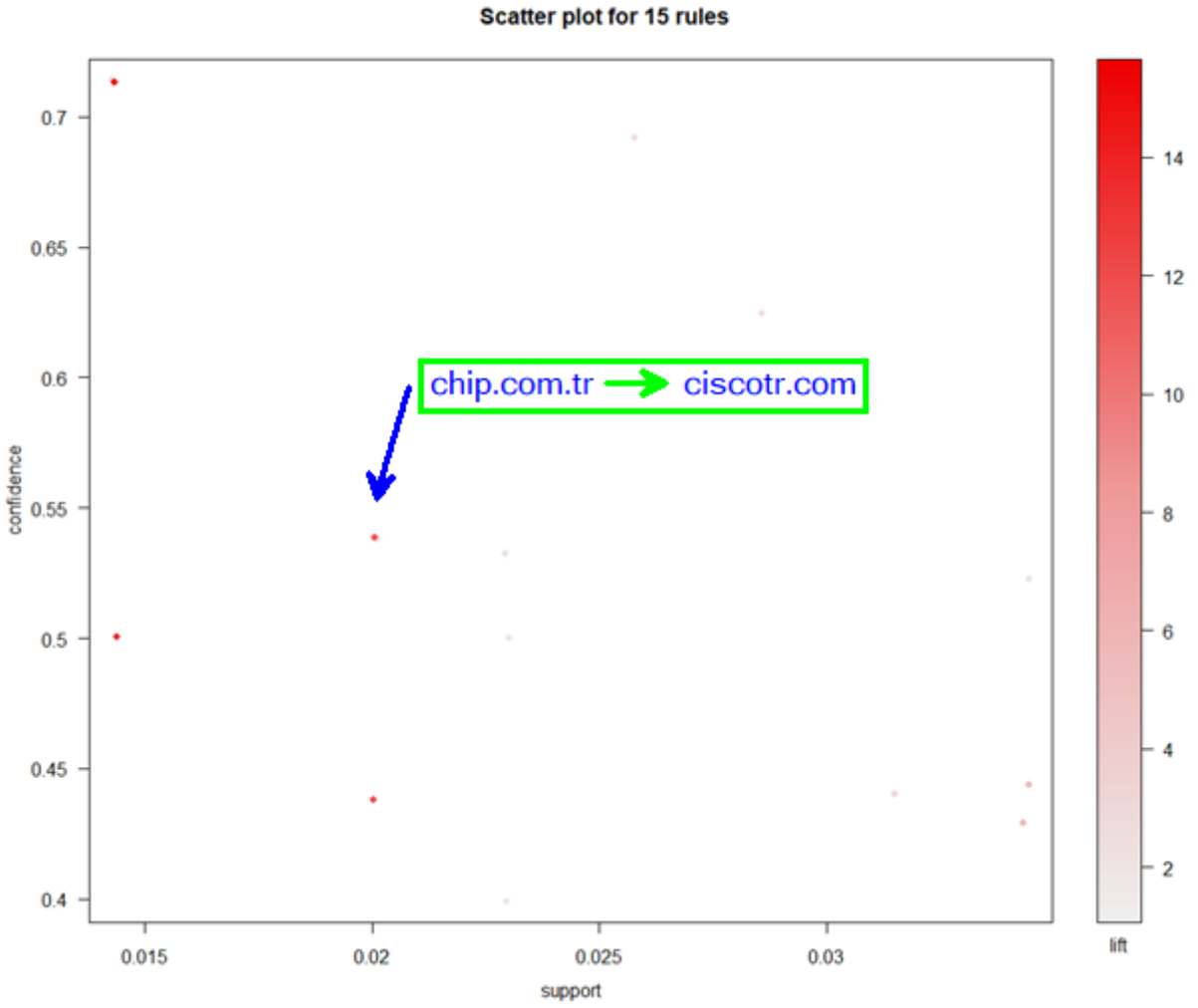
Eclat algoritması çalıştırıldığında da Şekil 4.52'de görüldüğü üzere apriori algoritması ile genellikle aynı ilişkiler görülmüştür. [9]



Şekil 4.52: R studio da eclat algoritması uygulaması sonucu elde edilen kural setleri

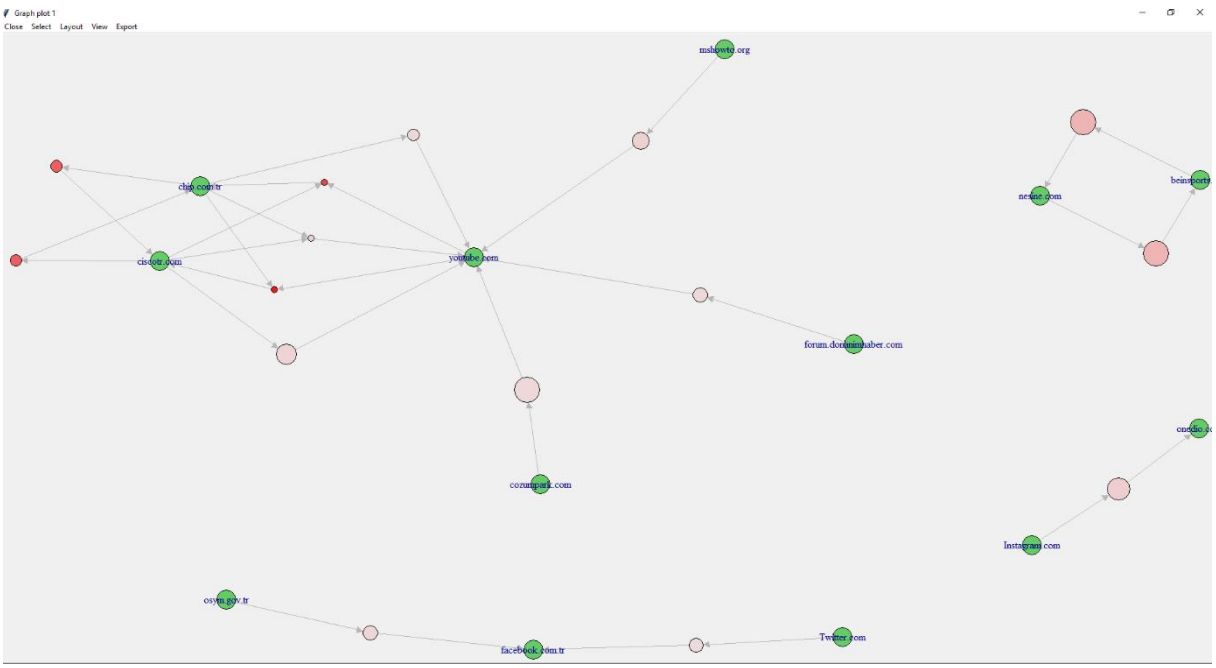
Apriori ile Eclat algoritmaları ilişki süreçlerini farklı algoritma arama yöntemiyle yapmaktadır. Apriori algoritması yayılım öncelikli arama yöntemi ile yaparken eclat algoritması derinlik öncelikli algoritmayı kullanmaktadır. Bu sebeple ilişkilerde kıyaslama yapıldığında ilk 24 ilişki ekranı alındığında örneğin aprioride 3 nolu ilişkinin eclatta 7 nolu ilişki olarak görünmektedir. Yine apriori-eclat eşleşmesi(4-6; 5-15; 7-22; 8-5 … vb) görünmektedir.İlişkilerin sıralaması değişik olsa da bulunan ilişki sonuçlarının aynı olduğu tespit edilmiştir.Yapılan çalışmada veri kümemiz büyüdükçe karşılaştırılan ilişki sonuçlarından ziyade performans-zaman kriteri değişmektedir. Yapımız büyüdükçe eclat algoritmasının verdiği cevap süresinin daha kısa olduğu tespit edilmiştir. [10]

Yukarıdaki sonuçlar kontrol edildiğinde laboratuvardaki gözlemlerle algoritmalarla alınan sonuçların yakın ilişkide olduğu görünmektedir. Örneğin apriori 9 nolu ilişki – eclat 4 nolu ilişkide görülen chip.com.tr ile youtube.com üzerinden proje ödevine destek alan öğrencilerin ciscotr.com.tr sitesine de gittiği görünmektedir. Bu ilişkinin güven değerinin %71.4 olduğu ve asansör değerinin 15.58'dir. Öğrencilerin chip ve youtube ile aldıkları verileri ciscotr sitesinden de desteklediği tespit edilmiştir.



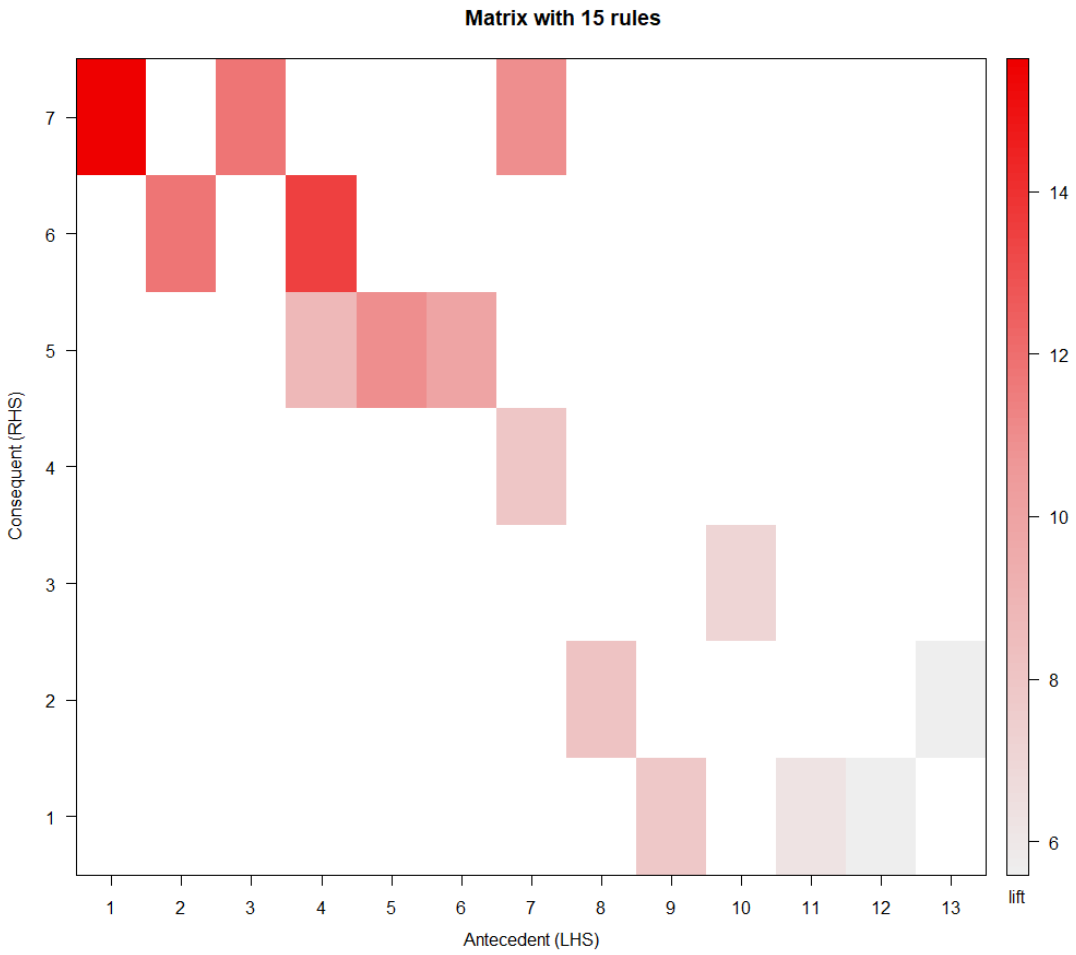
Şekil 4.53: Serpilme grafiği

Şekil 4.53'de veri setinden elde edilen 24 kuralın serpilme grafiği görülmektedir. Bu grafiğin x ekseni destek, y ekseni güven değerlerini belirtmektedir. Grafiğin x ve y eksenlerinin kesim noktasında yer alan noktaların renkleri kurallara ait asansör değerlerini göstermektedir. Güven değeri 0,5384615 , destek değeri 0,02 ve asansör değeri 11,74 olan kural chip.com.tr → ciscotr.com olarak şekilde de görülmektedir.



Şekil 4.54: Kural grafiği

Kural grafiklerinde, düğümler ve nesneler arasındaki ilişkiyi gösteren oklar kullanılmaktadır. Okların renk ve kalınlığına göre birliktelik kurallarının gücü ve yaygınlığı hakkında bilgi elde edilmektedir. Okların kalınlığı arttıkça kuralın desteği, renkleri koyulaştıkça da kaldıraç/güven değerlerinin arttığı anlaşılmaktadır. Şekil 4.54'da veri setinden elde edilen 24 birliktelik kuralına ait kural grafiği görüntülenmektedir.



Şekil 4.55: 2B matris tabanlı grafik

Şekil 4.55'de görüldüğü üzere grafiğin x ekseninde farklı öncül öge kümeleri, y ekseninde de aynı şekilde farklı sonuç öge kümeleri yer almaktadır. Öncül ve sonuç değerlerine sahip bir kuralın destek ve güven değerine göre grafikte bir hücre işaretlenmektedir. Grafikte minimum destek ve minimum güven eşik değerini sağlamayan birliktelik kuralları boş hücre olarak gösterilmektedir. Grafiğin sağ kısmında kuralların güven değerini gösteren kırmızının açık tonlarından koyu tonlarına doğru değişen renk skalası bulunmaktadır. Kurallara ait güven değerleri 0- 1 aralığında değişmektedir. Açık tonlu renkler düşük güven değerini, koyu renklere doğru yüksek güven değerini göstermektedir.

5. SONUÇ

Bireyler yaşam süreçlerini devam ettirebilmek için gerekli olan ihtiyaçları karşılamak zorundandırlar.Birçok insanın bir arada yaşaması bu ihtiyaçları bireysel karşılamasını imkansız hale getirmektedir. Bu sebeple makinalaşma ve makinalaşmaya bağlı bilişim süreçleri her geçen gün daha önem kazanmaktadır. Bireylerin yemek,giyim,tıbbı vb ihtiyaçlarını karşılamaları için teknoloji ve bilişim altyapısına ihtiyaçları vardır.Bilişim teknolojileri de her geçen gün artan bir ivmeyle hayatımızda yer almaktadır. Genel anlamda bakılırsa, bu yapıların çalışabilmesi için makine,ağ yapısı ve donanımsal yapı ile bu yapıların çalışmasını sağlayan kod,algoritma ve yazılımsal yapı olarak iki ana dala ayırabiliriz.

Çalışmamızda bu iki ana dal kullanılarak;ağ mimarisinden, kullanıcılarının ağ trafiği incelenip geçen veriler alınmış,yazılımsal mimari ile,veriler algoritma üzerinde işleme sokulup çıkan ilişki sonuçları tespit edilmiştir.

İstanbul Ayvansaray Üniversitesi / Plato Meslek Yüksekokulu / Internet ve Ağ Teknolojileri bölümünde okuyup 2018 yılında mezun olan öğrencilerin bir dönem içerisindeki seçilmiş 9 haftalık laboratuvar kullanımı kontrol edilmiştir.Laboratuvarda öğrencilerin oturduğu bilgisayarlar sabitlenip MAC adresleri üzerinden gittikleri web sitelerin IP adresleri ağdan alınmış, bu alınan veriler ile çalışmamızın veri seti oluşturulmuştur.

Verilerin anaconda derleyicisiyle, R studio derleyicisi üzerinde çalışması için gerekli düzenlemeler yapılmıştır. Öğrencilerin ders notu, cinsiyeti gibi özellkleri veri setine eklenerek çıkarılacak ilişkilerin boyutu arttırılmıştır. Verilerde öğrencilerin gittikleri web siteler listelenmiş ve bu siteler arasındaki ilişkiler apriori algoritması ile çıkarılmıştır. Apriori algoritması ile elde edilen bilgilerin doğruluğunu kıyaslamak için Eclat Algoritması ile ilişkiler kontrol edilmiştir. Apriori ve Eclat algoritmaları ilişkileri tarama yöntemi olarak farklı yöntemleri kullandıkları için tercih edilmiştir. Her iki algoritmada aynı sonuçların çıktığı gözlemlenmiş ve öğrencilerin cinsiyetleri arasında belirgin bir farklılık tespit edilmemiştir. Öğrencilerin ders notlarına göre derslerde gidilen sitelerin farklılaştığı gözlenmiştir. Bu sonuçlar ile öğrencilerin ders saatlerinde verilen uygulamaya paralel olarak bilgiye hangi siteden ulaştıkları da gözlenmiştir.

Yapılan çalışmada çıkarılan sonuç ve öneriler aşağıdaki gibidir.

- Düşük not ortalamasıyla mezun öğrencilerin özellikle sınav öncesi haftalarda daha çok bilişim konulu sitelere gittikleri tespit edilmiştir. Bu tespit konuların derste anlaşılmadığı veya dersi geçebilmek amacıyla kaynak taramasını internet üzerinden yaptıklarını göstermektedir.
- Öğrencilerin sınav haftaları hariç en çok sosyal medya içerikli sitelere gittikleri saptanmıştır.
- Yapılan çalışmanın kapsamı genişletilerek okulların bağımsız bilgisayar ortamlarında (kütüphane) uygulanarak, öğrencilerin ilgi ve tutum ölçekleri belirlenip okulun sosyal kültür ve sağlık birimlerine kaynak sağlanabilir. Okulda yapılması düşünülen etkinlikler önceden planlanabilir.
- Okuldaki öğrencilerin başarılarını etkileyen faktörlerin araştırılması için zemin oluşturulabilir.
- Bu çalışma uygun ortam sağlanırsa oluşturulan veri seti ile; öğrencilerin youtube gibi kanallardaki içeriklerin direk tespiti yapılarak daha gerçekçi ve detay verilere ulaşılabilir.
- Çalışmanın uygulanabilir olmasıyla bir kurumdaki kişilerin şiddet eğilimleri veya kurum için oluşturabilecek tehdit içerikli(yasaklı siteler) sitelere henüz gidilmeden tespit edilerek, bu durumun önüne geçilebilir.

KAYNAKÇA

[1] **P.Tan, M.Steinbach, A.Karpatne, V.Kumar.** Chapter 5: Assocation Analysis: Basic Concepts and Algorithms. *Introduction to Data Mining.* Minnesota : Pearson:2 edition, 2018.

[2] **S.Raschka.** rasbt.github.io. *http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/association_rules/.* [Çevrimiçi] MKDocs, 01 10 2018. [Alıntı Tarihi: 20 04 2019.] http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/association_rules/.

[3] *Analysing the quality of Association Rules by Computing and Interestingness Measures.* **J.Manimaran, T.Velmurugan.** 15, Tamil Nadu,India : Indian Journal of Science and Technology, 01 Temmuz 2015, Indian Journal of Science and Technology, Cilt Vol 8, s. 2-5. DOI:10.17485.

[4] **J.Han, M.Kamber,J.Pei.** *Data Mining / Concepts and Techniques - 3rd ed.* Waltham,USA : Morgan Kaufmann Publishers, 2012. ISBN: 978-0-12-381479-1.

[5] **R.Jain.** www.hackerearth.com Apriori algortihm for Data Mining. *hackerearth blog.* [Çevrimiçi] HackerEarth, 24 March 2017. [Alıntı Tarihi: 22 04 2019.] https://www.hackerearth.com/blog/machine-learning/beginners-tutorial-apriori-algorithm-data-mining-r-implementation/.

[6] **Admin, Techopedia.** Techopedia. *www.techopedia.com.* [Çevrimiçi] Techopedia, 01 01 2018. [Alıntı Tarihi: 20 04 2019.] https://www.techopedia.com/definition/30306/association-rule-mining .

[7] **K.J.Cios, W.Pedrycz,R.W.Swiniarski,L.Kurgan.** A Knowledge Discovery Approach. *Data Mining.* Verlag USA : Springer, 2007.

[8] *CRISP-DM: Towards a Standard Process Model for Data Mining.* **R.Wirth, J.Hipp.** Manchester,UK : R.Wirth, J.Hipp, 2000.

[9] *A Localized Algorithm for Parallel Association Mining.* **M.J.Zaki, S.Parthasarathy,W.Li.** Rochester,US : University of Rochester, 1997.

[10] *ECLAT Algorithm for Frequent Itemsets Generation.* **M.Kaur, U.Grag.** 03, Phagwara,Punjab,India : International Journal of Computer Systems, 2014, Cilt Vol 01. 2394-1065.

[11] *Uygulamalı Mekanik ve malzemeler.* **Dr.Zhi Gang Fang, Jian Jun Xu,Ping Wang.** 543-546, basım yeri bilinmiyor : Ris, BibTeX, Aralık 2014.